(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
16 January 2003 (16.01.2003)

PCT

(10) International Publication Number
WO 03/005247 A2

(51) International Patent Classification⁷: G06F 17/30

(21) International Application Number: PCT/US02/21235

(22) International Filing Date: 3 July 2002 (03.07.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/303,450 6 July 2001 (06.07.2001) US

(71) Applicant: COMPUTER ASSOCIATES THINK, INC. [US/US]; One Computer Associates Plaza, Islandia, NY 11749 (US).

(72) Inventor: VIVEK, Prakash; Computer Associates, One Computer Associates Plaza, Islandia, NY 11749 (US).

(74) Agent: JAWORSKI, Richard, F.; Cooper & Dunham LLP, 1185 Avenue of the Americas, New York, NY 10036 (US).

(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 03/005247 A2

(54) Title: SYSTEMS AND METHODS OF INFORMATION BACKUP

(57) **Abstract:** A system for backing up and restoring information includes at least one storage device, a primary server for granting and denying a global lock to devices requesting access to the at least one storage device and at least one secondary server including internal tables for storing information relating to storage in the at least one storage device, the at least one secondary server requesting a global lock from the primary server, wherein after the global lock is granted by the primary server, the at least one secondary server checks its own internal tables to determine whether access to the at least one storage device can be granted.

SYSTEMS AND METHODS OF INFORMATION BACKUP

BACKGROUND OF THE DISCLOSURE

The present disclosure claims the benefit of Provisional Application Serial No. 60/303,450, filed July 6, 2001, the entire contents of which are herein incorporated by reference.

1. Field of the Disclosure

The present disclosure relates generally to backup systems and in particular, to systems and methods for providing information backup.

2. Description of the Related Art

Backup storage devices are used for backing up information on servers and other network machines or devices. Backup storage devices are often used to backup various types of information including, for example, files, databases, system information and files, etc.

When one backup storage device is shared among a plurality of host devices, problems may arise when accessing the storage device to read or write to it. An example of a local area network (LAN) including backup storage is shown in Fig. 1A. The LAN includes one or more workstations 5a-5n, one or more servers 6a-6n and at least one backup storage device 7 interconnected via a network 9. Each of the workstations 5a-5n and servers 6a-6n my utilize different operating systems. Backup storage device 7 may be, for example, a tape device such as a tape library. If host devices such as the workstations 5a-5n or servers 6a-6n are sharing the same storage device 7, each operating system may attempt to store data in the same location on the storage device 7. In this case, data required by one host device might be overwritten by data from a subsequent write by another host device. In addition, some operating systems may store special information at specific addresses of the storage device 7. Some other operating systems may require that any attached storage devices have identifying information stored in one or more portions of that same area of storage. This results in conflicts and difficulties in providing such shared storage. A controller may be used to control access to the storage device 7. However, should the controller crash, the storage system data may be permanently lost .

Multiple storage devices may be provided for backing up data on host devices. However, managing multiple storage devices in an efficient and effective manner allowing backup and recovery for multiple hosts can be time and resource consuming and can utilize much of the bandwidth of the LAN or wide area network (WAN).

A storage area network (SAN) may be used to free up bandwidth on the LAN or WAN. A SAN is a high speed storage network of shared storage devices. A SAN makes all storage devices available to all hosts on the LAN or WAN, saving valuable bandwidth on the LAN or WAN.

Figure 1b depicts a SAN environment. The SAN includes a plurality of distributed servers 2a-2n, connected to one or more storage devices such as backup libraries 4a-4n, via a highspeed network 3. The backup devices 4a-4n may include, for example, tape libraries each including a plurality of storage media, a picker and one or more read/write devices for reading and writing to the storage media. The distributed servers 2a-2n and the backup devices 4a-4n may be interconnected via a high speed/high volume network 3 that may include various highspeed switches and/or bridges. With this type of environment, there are several ways that the servers 2a-2n and backup devices 4a-4n can be shared. For example, one of the servers 2a-2n may act as a chief arbitrator granting access of backup devices 4a-4n to all other servers. This is often referred to as a master/slave system. In a master/slave system, the intelligence lies with the master and the master has sole discretion and decision making authority regarding when and if to allow servers to access a backup device. The master creates and maintains information regarding the state of each of the devices on the network.

One problem with the master/slave system is that if the master crashes, it may take a lot of time and effort to configure another server as the master. All other servers will wait until another master server has been designated and configured by the system administrator. The new master will then re-inventory the network, and build up information regarding the state of each device on the network. This can be a time consuming and costly procedure.

Another approach is to partition access of a backup device among the servers, so that each server is capable of accessing only the drives and slots in the library within its partition. However, this system may not make the most efficient use of the backup devices.

In many backup operations, the backup application reads the data from the primary storage device to the server memory, does necessary formatting, and sends it to the appropriate backup storage devices. These processes involve heavy utilization of CPU, IO channels and the system memory. It is quite possible that when backup operations are in progress, the servers may not be able to perform any other application processing. This is one of the reasons why system administrators prefer to perform backup at a time when the application load on the server is at a minimum. It would be beneficial if the backup operations could be performed without requiring heavy utilization of the server system, thereby freeing up the server even during information backup.

SUMMARY

A system for backing up and restoring information comprises at least one storage device, a primary server for granting and denying a global lock to devices requesting access to the at least one storage device and at least one secondary server including internal tables for storing information relating to storage in the at least one storage device, the at least one secondary server requesting a global lock from the primary server, wherein after the global lock is granted by the primary server, the at least one secondary server checks its own internal tables to determine whether access to the at least one storage device can be granted.

If it is determined that access to the at least one storage device can be granted, the at least one secondary server may mark as locked, at least a portion of the at least one storage device and store corresponding information in its own internal tables and propagates the corresponding information to the primary server and any other secondary servers for storage in their own internal tables, indicating that the portion of the at least one storage device is locked. After the at least one secondary server propagates the corresponding information to the any other secondary servers, it may release the global lock so that the global lock is available to any of the secondary servers or the primary server. After the at least one secondary server releases the global lock, the at least one secondary server may begin backing up information into the at least a portion of the at least one storage device or restoring information from the at least a portion of the at least

one storage device. After the at least one secondary server is done backing up into or restoring information from the at least a portion of the at least one storage device, the at least one secondary server may again request the global lock so that it can update the primary server and any other secondary servers accordingly. After the at least one secondary server is again given the global lock, the at least one secondary server may send information to the primary server and any other secondary servers informing them to update their internal tables to unlock the at least a portion of the at least one storage device. The system may further comprise a network for connecting the at least one storage device, the primary server and the plurality of secondary servers. The network may comprise a storage area network.

Also disclosed is a method of granting and denying access to servers on a network for backing up information to and restoring information from a storage device, the servers including a primary server and secondary servers, comprising sending from a secondary server to the primary server a request for a global lock, the secondary server including internal tables for storing information relating to storage in the storage device, determining whether any other servers on the network have the global lock, and granting and denying the global lock to the secondary server based on whether any other servers on the network have the global lock.. When the global lock is granted by the primary server, the at least one secondary server checks its own internal tables to determine whether access to the at least one storage device can be granted.

If it is determined that access to the at least one storage device can be granted, the at least one secondary server may mark as locked, at least a portion of the at least one storage device and store corresponding information in its own internal tables and propagate the corresponding information to the primary server and any other secondary servers for storage in their own internal tables, indicating that the portion of the at least one storage device is locked. After the at least one secondary server propagates the corresponding information to the any other secondary servers, it may release the global lock so that the global lock is available to any of the secondary servers or the primary server. After the at least one secondary server releases the global lock, the at least one secondary server may begin backing up information into the at least a portion of the at least one storage device or restoring information from the at least a portion of the at least one storage device. After the at least one secondary server is done backing up into or restoring information from the at least a portion of the at least one storage device, the at least one secondary server again may request the global lock so that it can update the primary server and any other secondary servers accordingly. After the at least one secondary server is again given the global lock, the at least one secondary server may send information to the primary server and any other secondary servers informing them to update their internal tables to unlock the at least a portion of the at least one storage device. The network may comprise a storage area network.

Also disclosed is a computer recording medium including computer executable code for granting and denying access to servers on a network for backing up information to and restoring information from a storage device, the servers including a primary server and secondary servers, comprising code for sending from a secondary server to the primary server a request for a global lock, the secondary server including internal tables for storing information relating to storage in the storage device, code for determining whether any other servers on the network have the global lock and code for granting and denying the global lock to the secondary server based on whether any other servers on the network have the global lock. When the global lock is granted by the primary server, the at least one secondary server checks its own internal tables to determine whether access to the at least one storage device can be granted.

If it is determined that access to the at least one storage device can be granted, code may control the at least one secondary server to mark as locked, at least a portion of the at least one storage device and store corresponding information in internal tables and code may propagate the corresponding information to the primary server and any other secondary servers for storage in their own internal tables, indicating that the portion of the at least one storage device is locked.

After the at least one secondary server propagates the corresponding information to the any other secondary servers, it may release the global lock so that the global lock is available to any of the secondary servers or the primary server.

Code may be provided for controlling the at least one secondary server, so that after the at least one secondary server releases the global lock, the at least one secondary server begins backing up information into the at least a portion of the at least one storage device or restoring information from the at least a portion of the at least one storage device.

Code may be provided for controlling the at least one secondary server, so that after the at least one secondary server is done backing up into or restoring information from the at least a portion of the at least one storage device, the at least one secondary server again requests the global lock so that it can update the primary server and any other secondary servers accordingly.

Code may be provided for controlling the at least one secondary server, so that after the at least one secondary server is again given the global lock, the at least one secondary server sends information to the primary server and any other secondary servers informing them to update their internal tables to unlock the at least a portion of the at least one storage device.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete appreciation of the present disclosure and many of the attendant advantages thereof will be readily obtained as the same becomes better understood by reference to the following detailed description when considered in connection with the accompanying drawings, wherein:

Figure 1A is a block diagram of a LAN environment;

Figure 1B is a block diagram of a SAN environment;

Figure 2 is a block diagram illustrating an example of a network environment to which the present systems and methods may be applied;

Figure 3 is a block diagram illustrating an example of a SAN environment to which the present systems and methods may be applied;

Figures 4 and 5 are screen displays for allowing a user to configure the system;

Figure 6 is a block diagram illustrating a configured system;

Figures 7 - 12 are screen displays allowing a user to set up jobs in a job queue;

Figure 13 is a screen display allowing a user to backup a remote server;

Figure 14 is a block diagram showing various engines used by the servers of the present system and method;

Figure 15 is a flow chart for describing multiple stream backup according to an embodiment;

Figure 16 is a block diagram for describing a serverless backup;

Figures 17 - 19 are flow charts for describing various aspects of serverless backup;

Figure 20 is a diagram showing files and blocks of data for those files;

Figures 21 and 22 are GUIs allowing a user to configure libraries; and

Figures 23 - 25 are flow charts for describing serverless backup operations.

DETAILED DESCRIPTION

In describing preferred embodiments of the present disclosure illustrated in the drawings, specific terminology is employed for sake of clarity. However, the present disclosure is not intended to be limited to the specific terminology so selected and it is to be understood that each specific element includes all technical equivalents which operate in a similar manner.

The present system and method is capable of being implemented in a Local Area Network (LAN) environment as well as in various other types of environments including, for example, a Storage Area Network (SAN) environment.

Figure 2 illustrates the hardware and software environment in which the disclosed embodiments will be further described. The system includes one or more workstations 10a - 10n, one or more servers 11a-11n and one or more backup devices 13a-13n interconnected via one or more networks. The workstations 10a-10n and servers 11a-11n may user the same or different operating systems. The workstations 10a-10n may include one or more desktop computers, laptop computers, mainframes, etc. In the embodiment shown in Fig. 2, the network is a LAN environment, and the workstations 10a-10n, the servers 11a-11n and the backup systems 13a-13n are interconnected via a LAN 15.

In addition, or as an alternative to the above-described LAN environment, a SAN environment may be provided. In the SAN environment, the workstations 10a-10n may be connected to one or more of a plurality of servers 12a - 12n via a LAN as described above with respect to Fig. 2. However, in a SAN environment, servers 12a-12n are connected to one or more backup storage devices 14a - 14n via a network 16, as shown in Fig. 3. As will be described in more detail below, the servers 12a-12n may be assigned to server groups which include distributed servers, a primary server and backup storage devices. Network 16 is a high speed network that may include a system of switches and bridges forming a part of a fibre channel network. The SAN forms a high speed, high bandwidth system for communicating data. The high speed network 16 may be, for example, a fibre channel network, a shared SCSI system or an SSA type system.

Although only one server group will be described herein, a plurality of server groups may be provided on the SAN. The primary server controls the flow of information of all of the servers in that SAN server group. This prevents conflicts if two servers attempt to allocate a device or media at the same time.

Fibre channel adapters and fibre channel libraries may be connected to the network 16 and are referred to as nodes. Nodes may be connected to switch/hubs, creating a ring. The fibre channel ring, sometimes also called a fibre channel arbitrated loop, includes nodes connected by a fibre channel hub, so that each node can talk with each other node. Nodes log in to the ring when powered up or when first connected to the ring. This causes an event on the ring that all other nodes on the ring receive so that each is aware of new nodes. Each node on the ring is assigned a unique world-wide address or id by the manufacture, allowing each node to be identified when it logs onto the ring. Backup devices such as tape libraries connected to a hub use applications that are network-storage aware, so that they can access the devices without interfering with other applications also accessing the devices.

In the SAN environment described herein, software systems in accordance with the present disclosure are installed on the servers 12a-12n. The system is then configured to designate the primary and distributed servers and to assign each server to a group. For example, according to an embodiment, a display such as that shown in Fig. 4 is provided to a user. The display includes a panel 20 provided on the left hand side which provides a tree view of the configuration of primary and distributed servers. A panel 22 on the right hand side of the display shows all available servers having the present software installed and which have not yet been assigned to any server group. A user first selects the primary server, by moving the curser (not shown) and selecting one of the servers from the list on the right hand side of the display and then clicking a Make Primary button 24. The primary server may also be deselected by moving the curser and selecting the server from the left panel and clicking a Delete Primary button 26. After the primary server has been selected, the distributed servers are then selected and assigned to that primary server to form a server group. A distributed server is selected from the right hand window of the screen and the Assign Distributed button 28 is clicked. To remove a distributed server from a group, the server name in the left panel is selected and the Remove Distributed button 30 is clicked. This process is repeated until the server group is configured as desired by the user.

The storage devices are then assigned to the server group. This process is performed on the primary server. For example, to assign shared libraries to a server group, a Device Group Configuration dialog is selected from the primary server machine. This presents a window such as that shown in Fig. 5, which lists each of the server groups in the left hand panel 32 and available storage media devices in the right hand panel 34. Although the present disclosure will be described mainly by reference to tape libraries as the storage media devices, other types of storage media devices may be used as alternatives to or in addition to the tape libraries. Tape libraries not presently assigned to a group are displayed in the Available Devices right hand panel 34. To assign a tape library to a group, a library listed in the Available Devices panel 34 is clicked on and selected. The group in which the user wants to add the library to can then be highlighted in the left hand panel 32 by the user. The Assign button 36 is then clicked. The library will then be removed from the right hand panel 34 and placed beneath the device group in the left hand panel 32. Additional libraries can be added to the group by repeating the above steps. When finished, the OK button 38 is clicked.

An example of the thus configured server group is shown in Fig. 6. The network now includes a primary server 13, distributed servers 15a-15n and storage media devices 16a-16n.

Each server maintains a job queue of jobs that need to be performed. An example of a job queue is shown in Fig. 7. Clicking on the job queue tab 40, displays the job queue in window portion 42, as shown. The job queue identifies the server involved in the job, the job number, the job ID, the state of each job and the execution time for each job, as shown.

When a job is run or scheduled to run, it is submitted to the job queue by the user. The present system continuously scans the job queue for jobs that are ready to be executed. Jobs can be controlled by selecting a job from the job queue and right-clicking various control options. For example, the user can be given the option to change the job's status to HOLD or to READY (if it is currently on hold). HOLD signifies that the job is not scheduled to be executed. READY means that the job can be executed. The user can also quickly submit a job to the job queue by using a previously saved script. A script is a job that was saved to a file. The script contains the original source, destination, option, and schedule information for the job. The job can be modified to add options or additional sources to an existing job, without having to create a new job. The job can be rescheduled to quickly change a job's execution date, time, or status. This also allows a user to resubmit a Done job that is still in the Job Queue. For example, the user may want to do this if the job was not successful when it first ran. The user can also cancel an active job from the job queue and reschedule it for its next regular interval. A job can be cancelled and deleted from the job queue completely. Deleting a job will interrupt and remove the job completely from the job queue, and it will not be rescheduled. If the user then desires to reinstate the job, the user may recreate the job (if it was not previously saved to a script file).

A user can set up jobs to be performed by adding them to the job queue. A backup manager allows jobs to be added to the job queue. For example, the user can set the source, destination and scheduling information for each job. By specifying the source, the user can specify the items they wish to backup. By specifying the destination, the user can specify where they wish to have the source information backed up. By specifying the scheduling information, the user can specify when they wish to perform the backup. The backup manager includes a backup wizard, which provides the user with various windows for selecting options and specifying the job queue information. The source, destination and scheduling information can be specified by use of the backup wizard

For example, after the backup wizard is started, the user is requested to specify the source or server they wish to back up, by entering the Server name, user name and password. After the system validates this information, the user is presented with a window 100 such as that shown in

Fig. 8. As shown, the browser displays a machine tree 101 from which the user can select domains, machines, volumes (drives), shares, directories and files for backup. For example, to backup the entire server machine, the user highlights "My Computer" and then clicks Next button 104. The system then displays a window 106 such as that shown in Fig. 9, requesting the user to choose the backup destination by selecting the media for the backup. After selecting the media for backup, the user clicks Next button 104. The user is then presented with the backup options screen 108, as shown in Fig 10. This screen allows the user to customize the backup job by choosing the extent of the files that are to be backed up. For example, the user can select Full Backup 110 or Incremental Backup 112. If Incremental Backup is selected, only files that are new (or have changed since the previous backup) will be backed up. This screen also allows the user to set whether to backup with Verification 114 and/or Compression 116. After clicking Next button 104, the user can be presented with a window 118 as shown in Fig. 11. This window shows the information that was input and asks the user to verify the information. If information is incorrect, the user can go back and alter any of the entries by clicking Back button 103. If the user is satisfied with the information, the user clicks button 104 and is presented with a window 120 as shown in Fig. 12. Window 120 allows the user to select whether the backup job is to be Run Now 122 or whether it is be Scheduled 124 to run at a specified date and time. If Schedule button 124 is selected, the user is presented with a series of options to choose a start date/time, repeat method, repeat interval (e.g., month, day, hour, minute) and days in which that backup job is to be excluded (if any).

The present system thus provides an easy and intuitive way of scheduling backup jobs. The present system also allows remote servers/machines in another server group or not associated with any server group to be backed up from a local server. The remote machine should have an appropriate agent on it so that it can be identified by the local server, and so that the data to be backed up can be retrieved and backed up. The agent on the remote machine browses targeted directories and packages data to be transported in pulses of large data packets to the local server. The local server then prepares the data to be stored on the designated storage devices.

Each agent may also include an access control list (ACL) identifying which local servers are able to backup or restore data through that agent. The access control list can identify specific users and/or IT addresses which are denied or allowed to backup/restore data through the agent.

The present system provides a security system which relies on the user having logon credentials on the node that the agent is deployed on. The connection from the local server to the machine having the agent (agent node) is established only if the user has login access to that agent node. The present system also provides an added level of security since the user should have physical access to the agent node (over and above the login access) in order to successfully operate on that node.

The ACL defines for a particular agent node the system servers (on the enterprise network) that can connect to it and access its functionality. The functionality is set on the node which hosts the agent and requires physical local node access in order to be manipulated. The ACL allows the user of that node to either set an include list or an exclude list of servers that can access it. By default the ACL can be set to grant all servers access to the Agent.

In order to change the ACL of a particular agent, the user locally logs in to that node and manipulates the ACL via an Agent admin utility. Users can invoke the agent admin utility and add or remove access requests via an Access Control List Dialog.

An example of adding a remote server for backup will now be explained by reference to Fig. 13. A backup manager screen 130 is presented to the user of the local server. In this example, the remote server to be backed up is a server that is running Windows NT. The user highlights the "Windows NT/Windows 2000 System" from window 132 as shown, and clicks on it. The user is then presented with the option to "Add Machine" or "Add Using Auto-Discovery", as shown. Clicking on Auto-Discovery will find machines that have the appropriate agents on them. For example, in this example, if Auto Discovery is selected, the system will find machines on the network that have the Windows NT/Windows 2000 system agents on them. Selecting the Add Machine as shown, the user is presented with a series of dialog boxes, allowing the user to enter a host name for the remote server/machine, and asking for a user name and password entitling the user to browse the remote server/machine and perform backups on it and allowing the user to set the destination for the backup.

Each server of the present system may include engines for performing various functions for backup, restore, etc. For example, as shown in Fig. 14, each server may include a Job engine 136, a Tape engine 138 and a Database engine 140. These engines perform the workload for the present system, processing and monitoring backup and restore operations and updating information regarding performance in activity logs and databases.

The job engine processes jobs in the job queue at the designated dates and times. The job engine scans the job queue for a job that is ready to run, and then sends it to the appropriate handler. Each server running the job engine can be configured to suit each user's needs. The job engine thus controls the execution time of jobs in the job queue. It scans the job queue regularly, launching jobs as their execution dates and times are reached. Various features of the job engines can be user controlled. By choosing a select configuration menu, for example, various options can be configured. For example, as noted above, the job engine constantly scans the job queue for jobs that should execute. By default, the job queue is scanned every 10 seconds. To change this time interval, the user can specify a time interval number from 1 - 999 seconds. The retention time for done jobs can also be controlled. That is, jobs with a final status of "DONE" will remain in the job queue for the time specified in this field. By default, DONE jobs are kept for 24 hours before they are deleted from the queue, unless they are recurring jobs. To change this time, a number between 0 and 999 can be input.

The system may also include an Activity Log which can be viewed by the user. The Activity Log contains information about all of the system's activities. By default, notes, warnings, and errors that occur when running the present system appear in the Activity Log. To change the types of messages that can be maintained in the Activity Log, the user can specify the following values: " None" so that no messages appear; "Errors" so that only errors that occur while running the present system will appear; "Warnings & Errors" so that warnings and errors that occur while running the present system will appear; "Notes, Warnings & Errors" (which is the default value) so that all notes, warnings, and errors that occur while running the present system will appear; "Debug" so that debug information appears which can be used for troubleshooting purposes.

The tape engine is responsible for communicating with and controlling the storage devices. The tape engine selects the device needed for a job. That is, the tape engine identifies all the storage devices that are connected to the system and activates them when a job is started.

Changing the configuration of the tape engine is not recommended. Normally, a default configuration is used, except for troubleshooting purposes. However, the present system provides various tape engine options. If a "Level" option is kept at the default ("None"), the user does not need to specify anything else. However, if desired, the following values are available: "None" (default) in which no information is logged. If the default is kept, a Tape Engine Log will not appear in a Job Status Manager. If set to "Detail," each message sent by the storage devices will be logged, with the system monitoring the tape engine and logging all information received. If an "Engine Log" option is selected, the Tape Engine Log will appear in the Job Status Manager. If "Output" is specified and "Detail" is set as noted above, the user can specify where they want the messages sent. The user can specify one of the following: "Both Screen and File" in which case the messages are recorded in the Tape Engine Log as well as being sent to a DOS box (a Tape Engine Message window). If "Screen Only" is specified, the messages are sent to the Tape Engine Message window only. If "File Only" is specified, the messages are recorded in the Tape Engine Log only. The user can view the Tape Engine Log in the Job Status Manager.

All operations performed within the system are recorded by the Database engine. The Database engine is responsible for maintaining a history of files, directories, drives, and machines that the system has backed up or copied. The Database engine is also responsible for maintaining information about the jobs that have been processed by the system, such as the job type, logs, the final result and the start and stop times of the jobs. The Database engine also maintains a history of the media used by the system, such as its type, its name, the date it was first formatted, the date it expires and the sessions on it. The Database Engine stores statistical information for all jobs processed. Some of the job information saved includes what files and directories have been backed up, copied, and restored, jobs that the present system has processed, storage devices and the media used for the jobs.

Various database engine options can be user controlled via a user interface. For example, a "Minimum Free Disk Space Required (M Bytes)" option allows the user to specify the minimum amount of disk space (in megabytes) the user wants to reserve for the system database. By default, 5 MB of disk space is specified. To change the amount of disk space required, the user can specify a number between 1 and 10. An "Enable Database Pruning" option can be set so that information regarding the files and directories that were backed up or copied in a session can be deleted when database pruning is on. By default, this option is selected, in order to free up space in the database file. However, the user may find it useful to set this option off in order to maintain the detailed information, for restoring purposes at a later time. However, it should be noted that not pruning the database can mean that the database may fill up quickly. A "Prune Database Records Older Than _ Day(s)" option is active only if the Enable Database Pruning option is on. The user can specify how long the record should be in the database before it is deleted. By default, the record will be deleted (if enabled) after it has been in the database for 30 days. To change the length of time, the user can specify a number between 1and 365. A "Run Database Pruning at _" option is active only if the "Enable Database Pruning option" is on. This option allows the user to specify when the pruning operation is to be run. By default, pruning (if enabled) will occur at 12:00 AM. A "Delete Media-Related database Records ONLY when Pruning" option may also be provided. When the user reformats or erases a media, the present system will delete the records in the database that pertain to the tape. Performing this extra step, however, can be time-consuming process. Accordingly, the user can select the "Delete Media-Related database Records ONLY when Pruning" option to postpone deleting these records until pruning is performed. A "Submit Database Pruning now" option allows the user to immediately submit a database pruning job to the job queue with the specified parameters. An "Enable Remote Database" option enables the server to use a remote database as its database, regardless of the remote database type. This database can be used to submit a restore job to a local machine. Configuration of the remote database is performed here as well. The user enters the Server Name, Login Name, Password and Server Operating System type of the remote database the user wants to enable.

The present system may also include several other types of managers for managing the system. A Restore manager restores data that has already been backed up by the present system. A user can set policies to schedule and manage the restore operations. The Restore manager can perform a complete or partial restore of data.

A Database manager displays information from the database, such as the jobs processed by the present system, the media used, and the devices that are being used by the system.

A Job Status manager monitors all pending, completed and active jobs from a Job Status Manager window. The Job Status manager also reschedules pending or completed jobs, submits new jobs, deletes jobs and stops active jobs in response to a user request. Log information is provided for each completed job.

A Devices manager displays information about the storage devices and media. The Devices manager allows a user to change a drive's compression mode, and perform media functions such as compression, formatting, erasing, ejecting and retensioning.

A Report manager generates reports from data in the database. Various reports include information about schedule policies, media errors, sessions, backup devices and media policies.

A Media Pool manager manages, creates and maintains logical groupings of media for easy identification of backups, allowing efficient scheduling of the maintenance and recycling of the media. The user can design media rotation schemes to suit the particular archive needs. Reports detail the media information for media pools used in the rotation schemes, including media status and session information.

An Alert manager sends messages to people in an organization using various methods of communication, regarding events that occur during the functioning of the present system.

Each time a backup job is run, the system records information in its database about the machines, directories and files that have been backed up, and the media that was used. This allows a user to locate files whenever the user needs to restore them. The database information is backed up by default whenever a user backs up the systems home directory. A user can select to backup the database files in every backup job regardless of whether the system home directory is selected for backup.

The present system also offers several utilities that can be used to manage files. For example, database recovery can be performed, should the data in the database be corrupted for any reason. A Recover Database utility lets a user restore information from a database backup to the database, overwriting existing database files. A user can use this utility any time the database files are lost, such as during a disaster like a fire, flood or machine crash. Several advanced database recovery options are available, including an option to specify a date to restore the database. In this case, the database session created on or before this date will be used for backup.

Another option allows the user to specify media options for the job. Another option allows the user to run commands or batch files before and/or after the restore job. An option is also provided allowing the user to determine the level of detail to record in the Job Queue Log.

A merge utility may be provided, allowing information to be inserted into empty slots within the database. This function allows a user to restore data from a different server than the one where the database resides. For example, if a database backup was created using the present system on a different machine, the user can use the Merge utility to get the media information into the database in the systems home directory. The Merge utility thus allows a user to take a media that contains one or more backup sessions and append the information from that media to an existing database. This is useful if detail information has been pruned from the database. Advanced merge options include an option to specify media options for the job, an option to run commands or batch files before and/or after the job, an option to determine the level of detail the user wants recorded in the Job Queue Log and an option to send messages about events in the operation.

A Scan function provides information about any media backup sessions. Each source that is selected to back up is saved on media as an individual session. A user can scan a single session or an entire media. Results of a scan job can be seen in the Job Queue. This function is particularly useful if a user is trying to recover a server and needs to find the most recent backup of the database so that it can be restored. A user can also use the Scan function if they want a list of the files that were backed up. Advanced scan options include an option to specify media options for the job, an option to specify some general options for the job such as to scan files with CRC verification, or to enable database recording, an option to run commands or batch files before and/or after the job, an option to determine the level of detail the user wants recorded in the Job Queue Log and an option to send messages about events in the operation.

A compare function compares the contents of a media session to files on a machine. Results of the Compare job can also be seen in the job queue. A user can use this feature after a backup to verify that the backup copied all of the files to media without error. Advanced compare options include an option to specify media options for the job, an option to specify whether or not to enable database recording, an option to run commands or batch files before and/or after the job, an option to determine the level of detail the user wants recorded in the Job Queue Log and an option to send messages about events in the operation.

A count function counts the number of files and directories on a machine. The results of the count function can be seen in the job queue. This option can be used after a copy job to verify that the copy function copied all of the files from one disk to another without error. Advanced count options include an option to specify whether or not to enable database recording, an option to run commands or batch files before and/or after the job, an option to determine the level of detail the user wants recorded in the Job Queue Log, an option to scan files for viruses before they are counted and an option to send messages about events in the operation.

A purge function allows a user to delete files and directories from a machine. Results can be seen in the job queue. The purge function allows a user to delete files and directories from a machine. Advanced purge options include an option to specify some general options for the job such as to remove directories or enable database recording, an option to run commands or batch files before and/or after the job, an option to determine the level of detail to record in the Job Queue Log and an option to send messages about events in the operation.

A copy function allows a user to copy files from one location to another. For example, a user can run a copy job on a local machine to store files and directories on another machine that is going to be backed up to media. Copy options determine related actions that occur during or after the copy operation. For example, an option is provided to specify when to retry open files and file sharing options. An option is also provided to specify whether or not to enable database recording. Another option creates a mirror image of the source machine on the destination media. An option is provided to copy files and directories from the source machine delete all files and directories that do not appear on the source machine. An option is also provided to delete files from the source machine after copying to the destination machine. An option is available to copy the file-level security information for file access on NTFS volumes. An option is also provided to run commands or batch files before and/or after the job is run. An option allows the user to determine the detail they want recorded in the Job Queue Log. Another option scans files for viruses before they are copied. Another option sends messages about events in the operation.

The present system allows data to be backed up by running storage operations in parallel utilizing multiple streaming. That is, the present system can be arranged to create multiple streams of backup for a backup job, based on the availability of tape devices, automatically or in response to a user selected option. By distributing the data across parallel streams, this approach greatly reduces the time required for backup operations. Recovery after a catastrophic loss will also be faster, since all available devices collaborate to restore all or most of the backup data at once.

Multiple streaming thus allows users to take advantage of all the available tape devices on the system, by splitting a single backup job into multiple jobs using two or more of the tape devices. As a result, multiple streaming increases the overall backup throughput compared with sequential methods. If more than one drive and more than one volume in the system are available, the user can choose to utilize multiple streaming.

Multiple streaming works for both local and remote devices. Backing up a remote source using the multiple streaming option is supported at the node (computer) level. A user can use all of the storage media devices, or can specify only a single group of storage media devices (if there are mixed types of storage media devices in the system). If a library option is installed on the system (as will be described below), multiple streaming (when selected) will use all of the available library devices. If the library option is not installed, the user may put the storage media devices into separate groups.

The total number of streams (slave jobs) created will depend on a number of storage media devices available, if it is a changer, otherwise, it will depend on the number of storage media device groups.

Multiple streaming is performed at the volume level for regular files (two volumes can run simultaneously on two separate devices), and at the database level for local database servers. Multiple Streaming is performed at node level for Preferred Shared Folder, remote database servers and Windows NT/2000 Client Agent.

The system can have as many jobs running simultaneously as the number of devices/groups that are on the system. When a multiple streaming option is selected, when one master job is created by the user, it will trigger slave jobs for as many volumes or devices of storage media as are available. When a job is finished on one volume, another job is executed until there are no more jobs to run. .

When utilizing multiple streaming, the user may wish to select a media pool, to prevent tapes from being overwritten. Separate tape devices should be configured in separate groups for regular drives. For tape systems with changers, they can be configured to be in the same group. Canceling a master job, implies canceling all the slave jobs. The user can cancel the master job or any one of the slave jobs to cancel the whole multiple streaming job completely. As described herein, when a job is put into the job queue, operations may be performed before or after the job is performed. For example, messages may be sent to various personnel, etc. These are referred to as pre/post operations. The pre/post operations and comments are supported at the master job level. That is, pre/post operations will not be performed for each slave job. Instead, the pre/post operations are performed before or after the complete job that spawned the slave jobs is run Each slave job will have a default job description in the job queue with this pattern: MULTI STREAM [JOB [MID]] SUBJOB [SID], where MID is the master job id and the SID is the sequence id. The Multi Stream option will be ignored if the system has only one storage media device. In order to achieve the optimum performance with multiple steaming jobs, a high-end server machine with multiple processors and at least 256 MB memory per processor may be utilized.

Fig. 15 is a flow chart for describing a multistream back up method according to an embodiment of the present disclosure. In Step S2, the user creates a master back up job. In step S4, the job engine scans the job queue, looking for jobs that are ready to run based on their scheduled run dates and times. In Step S6, a determination is made whether there is a job to run, based on the scheduled start times. If it is not time for a master job to run (No, Step S6), the job engine continues scanning the job queue until it is time for the master job to run (Yes, Step S6). In Step S8, the system splits the master job into multiple smaller slave jobs, with each slave job corresponding to a backup/restore of a single source entity (file system, database, etc.). The system then sends the smaller slave jobs to the job queue (Step S10). In Step S12, the job engine continues scanning the job queue for jobs. This continues until the new slave job is detected (Yes, Step S14). When the new slave job is detected, the job engine communicates with the tape engine to determine if a tape device is available to begin backing up to (No loop, Step S16). This continues until a tape device is determined to be available. Once a tape device is determined to be available (Yes, Step S16), the smaller slave jobs are started for each job having an available storage device (Step S18). As each slave job completes, information is returned back to the tape engine to indicate that the tape device used by the slave job is again available (Step S20). The job engine then scans the job queue for the next slave job and the process repeats until all slave jobs have been performed. A determination is made whether all jobs have been completed. If not completed (No, Step S20), the process again scans the job queue looking for a slave job to perform (Step S12). If each slave job is complete (Yes, Step S20), the process ends (Step S22) and the master job is deemed complete. At this time, any post job operations may be performed.

The present system is also capable of performing serverless backup, which solves many traditional backup related issues for system administrators. Serverless backup allows a user to perform backups with almost zero impact to the system CPU and allows applications on servers to continue to run while the backup is in progress. Serverless backup eliminates the need for a backup window, giving the highest possible backup/restore performance and allowing applications to continue to access the data while the backup is being performed.

Serverless backup is implemented using the industry standard extended copy command, and supports SAN environments with devices that support the extended copy command.

By using the extended copy command, serverless backup effectively eliminates the server completly from the data backup transfer path, and achieves direct disk to tape or tape to disk data transfer. Initiating the data transfer is the only activity that happens at the server level, with the actual data transfer happening at the Data Mover level on the SAN. The serverless backup according to the present disclosure transfers data between disks and tapes using "Data Movers." Data Movers are devices that support the extended copy command, eliminating the need for bringing the data all the way to the server, achieving the shortest data path. This enhances the backup performance considerably, and yields the highest possible backup/restore speeds. This allows the applications to continue to access the servers while the backup is in progress. This functionality eliminates the need for a backup window, and allows systems administrators to perform a backup at any time.

An example of a system for serverless backup is shown in Fig. 16. As shown, the system includes Servers 150a-150n, disk array 152 and backup storage media 154 which are all interconnected on the SAN via a high speed network 160 which may include routers, bridges, etc. Disk array 152 is capable of storing data from various workstations (not shown).

The SAN supports "extended copy" in which the data can be sent directly from disk array 152 to backup storage media 154, without having to go through a server. As shown by arrow A in Fig. 16, the data path for the backup of data starts at the disk array 152, travels through the network 160 and arrives at the backup storage media 154. Since the data path does not include the use of a server, the backup is considered serverless.

The only way that the servers are involved are to receive data regarding the backup. For example, the servers may communicate with various portions of the system to get metadata about the backup, etc. The servers may also be used to start and control the backup process, but the data being backed up moves directly from disk array 152 to backup storage media 154 without going through any server's CPU.

One aspect of the serverless backup is based on parsing file-mapping tables of each volume. Fig. 23 is a flow chart for describing aspects of serverless backup according to an embodiment of the present disclosure. The system first opens up the NT file system (NTFS) root directory and structure (Step S70), parses the same for allocation tables of each file (Step S72), and finds out attributes of each file (Step S74). Upon examination of the attributes of each file, the system determines whether the file is resident or non resident (Step S76). If the file is resident (Yes, Step S76), entire attributes are backed up (Step S80). If not resident (No, Step S76) , the system backs up the file attribute (or attributes), as well as the data blocks that belong to the file (Step S78), thus mapping the entire volume for all files residing on the volume for complete data protection. On a system supporting file allocation table (FAT) file systems, the system can translate the blocks by translating the FAT schema. The resulting scheme is a Serverless backup using extended copy technology capable of providing file level restores.

As shown in Fig. 24, for NTFS, the present system opens a volume (Step S82), reads in the boot sector (Step S82)and gets the bios parameter block (BPB) (Step S86). The system then determines the sector size, cluster size, file record segment size, clusters per Index Allocation Buffer (Step S88).

The system traverses the root directory first (based on the file reference of root directory, the system reads the file record segment for root directory), depending on the index root entry (Step S90) and determines if it is a file (Step S92). If it is a file (Yes, Step S92), the system will examine the attribute of the file (security, data attribute etc), and backup the corresponding value of the attribute. A determination is made whether the file is resident or nonresident (Step S96). If the value is a resident value (Yes, Step S96), the system will backup the file record segment containing the attribute. If it is non resident (No, Step S98), the system will figure out the LCN/VCN (runlist) and remember the information (Step S100).

If not a file (No, Step S92), other steps are performed (Step S102). For example, if the root entry is index-node, the system will look into index allocation attributes. If it is a directory, the system will traverse through the directory. The system will then add all other clusters, which are not in the list yet, based on a bitmap.

As shown in Fig. 25, for FAT/FAT32, the system opens the Volume (Step S104), reads in the boot sector (Step S106) and gets the BPB (Step S108). The system reads in the root directory (Step S110), and based on each entry in the root, the system may parse it differently. For example, if it is a sub directory(Step S112), the system will determine the cluster chain of the directory (Step S114), read in the subdirectory (Step S116) and traverse through it (Step S118). If it is a file (Step S112), the system will determine the cluster chain of the file using a FAT table (Step S120)

For a snapshot, before backup, the system flushes the volume's cache. The system then freezes the volume (e.g., takes a snapshot) During a backup analysis stage, a snapshot driver will return the original data to a Serverless/image, for the actual data backup. For the serverless system, since data movement happens to the raw disk, a data mover may not be able to give back the original data. Accordingly, at the end of serverless backup, the system may ask the snapshot driver to give all the original data of the modified clusters, and the system will apply that information during restore.

Another example of a system flow for performing serverless backup is shown in Fig. 17. To accomplish serverless backup, the present system obtains a static view of the disk array 152 or portion of the array (Step S40), maps the blocks of data on the disk set to the files to which they belong (Step S42) and moves the data directly from disk array 152 to backup storage media 154 (Step S44).

In order to obtain a static view of the data in the disk array, the present system may use a backup mirror system. An example of a flow for obtaining a static view of the data in the disk array using a backup mirror system is shown in Fig. 18. First, the mirror is established (Step S46). Next, any applications writing to the disk array are stopped (Step S48) and the mirror is then split (Step S50).

Another way to obtain a static copy is to use a snapshot. In this case, as shown in Fig. 19, the system first stops any applications writing to the disk (Step S52) and then creates the snapshot (Step S54).

The system now has a static view of the disk array to back up. When the data is being transferred from the disk array 152 to the backup storage media 154, the data may be transferred as blocks of data as opposed to being transferred as files. Accordingly, in order to eventually restore a file, the system creates a table or map identifying which blocks belong to which files. For example, as shown in Fig. 20, File A consists of data blocks A, B, and C. File B consists of data blocks D, E, and F. When the snapshot or split mirror is created, these mappings will remain constant until another split mirror or snapshot is created. The present system creates the table or map recording the data block associations for File A and File B. When the user then attempts to restore File A, for example, the present system looks up the blocks associated with File A and restores them. That data blocks being sent from the disk array 152 to the backup storage media 154 are directly transferred using the extended copy command.

When restoring data from the server-free backup system, the user has the option to restore the data on the image level for restoring the entire drive or on the file level for recovering individual files or directories.

An image level restore copies the entire image back from the backup storage media 154 to the disk array 152. File level restore acts just like nonserver-free restores, copying the selected file(s) from the backup storage media 154 to the disk array 152.

To perform image-level server-free restores, the present system creates a snapshot or split mirror to which the data is going to be restored. Next, the present system uses the extended copy command to restore the backed up data directly from the backup storage media 154 to disk 152.

In the above-described embodiments, any applications writing to the drives are stopped while a snapshot or image of the drive is taken. However, instead of stopping the applications writing to the drive, the present system is also capable of waiting for a preferred period in which to take the snapshot or image. For example, the system can synchronize the drive 152 and wait for a period of write inactivity to the drive 152. By default, the period of write inactivity the system looks for prior to freezing the drive (e.g., taking the snapshot), is 3 seconds. However, this period can be user modified if desired. If the specified period of write inactivity (3 seconds) cannot be found, the synchronization times out after a specified timeout period (e.g., 80 seconds) and the freeze request will fail. The specified period of write inactivity is the number of seconds of inactivity what should occur on the drive before the drive is deemed safe to freeze. The timeout period can also be user modified if desired. After the drive is deemed safe to freeze, the snap shot or image of the drive can be performed. In this way, applications are not forced to stop in the middle of processes they may be performing.

The present system may also include GUIs for managing one or more tape libraries. A library is a device containing one or more tape drives with an automated media delivery system, such as a robotic picker, which allows large amounts of data to be backed up without manual intervention. A library may include various components including Barcode ID (if the library supports it). The present system can user the barcode ID to quickly locate individual media by reading the barcode affixed to the media. The libarary may also include one or more magazines (or bins), where one or more groups of storage media (such as tape media) are stored. A robotic picker may be provided that moves the storage media units between the magazine slots and the drives. The library may include one or more drives in a device that transfers (backs up or restores) data between the computer or server and the storage media. A library can contain one or more drives, and from one to several hundred magazines. Smaller libraries are sometimes referred to as autoloaders, or changers. As used herein, the term "library" is used to refer to libraries of any size.

A library can be considered similar to a single storage drive. However, one of the primary differences is that a library typically uses at least two SCSI IDs: one for the library itself and one for each of the library's drives. Some libraries may use a single SCSI ID. When a library uses only one SCSI ID, the library and the library drive may each be assigned a Logical Unit Number (LUN) to differentiate between the drive and library. A single SCSI ID is, in effect, split into two Ids. Using a library has several advantages over the user of individual storage units. For example, using a library, no manual intervention is necessary during backup and restore operations. In addition, libraries can contain multiple storage drives and a large number of media, thus providing large capacity capabilities.

The present system is capable of supporting libraries with multiple drives, as well as those with single drives. The present system can also support multiple libraries. The present system allows a user to track and control an initialization process of the libraries from one location. For multiple drive libraries, the tape engine uses all of the drives for the initialization process. The present system also allows for multiple, concurrent, device management functions, including storage drive cleaning from any specified slot. Support is also provided for various library device management functions including format slot range, erase slot range (both quick and long), import/export, inventory slots, mount magazine, dismount magazine, clean library drive. The present system is capable of using all available drives on a library. If multiple drives are available, device management functions can be performed concurrently.

The present system can quickly locate individual media by reading the barcode that can be affixed to the media. After installation is complete and tape engine is started for the first time, the entire library is inventoried and recorded. The present system provides device configuration which allows the scheduling of library cleaning, providing the library supports this feature. Libraries can be grouped. Library groups employ the same concept as device groups (e.g., server groups) except that library groups are based upon slots. Any of the slots inside a library can be selected to form a library group. If the library runs out of tapes to span to, the user can insert a new tape, and the present system will be able to identify the tape as part of the pre-defined group. The present system can inventory new media which allows a user to insert new media into a library and then continue the backup or restore operation without interruption. If a drive in a multiple drive library becomes defective, the defective drive can be marked as offline. This enables the Tape Engine to continue to operate using the remaining non-defective drives in the library. The present system can search for a media by name or by a slot number depending on the user's preference. The present system provides large library support, allowing a changer with more than 1000 slots to be displayed in a list to the user. Slots can be displayed in four ways: large icons, small icons, details, and list.

The library can be configured from a Device Configuration menu. First, a tape library is chosen from a Device Configuration menu. A library setup screen then appears. If the tape engine is running, a window appears to let the user stop the engine. The tape engine should be stopped in order to configure a library device. If the library supports device serial numbers, the drives should already be configured and will appear in a Library Devices list. If the drives have not been assigned to the Library automatically, they can be assigned manually. This can be done by first highlighting the drive and the Library, and then clicking an Assign button. Assigning a drive to a library allows the present system to be aware of the drive's existence within the library. If drives are being assigned manually, the drives inside multi-drive libraries should be assigned in the correct order. For example, usually the library is configured by the manufacturer in such a way that the first library drive has the lowest SCSI ID number, and the last library drive has the highest SCSI ID number. However, this may not always be the case.

The next setup window then appears. The user highlights the library and clicks a Properties button so that a properties dialog appears. The Properties dialog allows the user to specify which slot holds the cleaning tape (if present) and if a bar code reader is installed.

After configuring the library, the present system runs an inventory. At this stage, the user can mark drives as either online or offline. In this way, defective drives (in a multi-drive library) can be marked as being offline, so that the present system can continue to operate using the remaining non-defective drives. When the drive is repaired or replaced, it can then easily be brought back online, by again performing the library setup procedure described above.

The present system also includes a virtual library feature for addressing a wide spectrum of storage requirements. The virtual library feature enables physical libraries to be configured into smaller virtual (logical) libraries, while still sharing the same robotics, import/export slots and cleaning media. The Virtual Library feature provides the flexibility to configure large libraries into smaller virtual libraries enabling a user to create different combinations of RAID level devices and/or stand alone tape libraries.

A Virtual Library configuration utility is used to set up all virtual libraries. This utility allows the user to define virtual libraries and their parameters, including the number of slots and drives required for the library. At minimum, a virtual library should have at least one slot and one drive associated with it.

The user can create a virtual library using the following steps. First, the user selects a Device Configuration menu. After clicking a Next button, a Device Configuration dialog is displayed. The user can select Virtual Library and click the Next button. A screen is then displayed, allowing the user to select from an Available Library window, the library to be configured into a virtual library. After selected, the characteristics associated with the parent physical library are displayed in a Detail field. Information associated with an adapter card, SCSI Bus, SCSI ID and LUN is displayed. In addition, a field displays the total slots available for this virtual library. From a library tree that is displayed, the user can select the drives to be associated with the newly created virtual library. Drives can be assigned to a virtual library in any order, although

drives cannot be shared between virtual libraries. In addition, each virtual library should have at least one drive and one slot assigned to it. The user then selects the number of slots to be assigned to this virtual library. The newly created virtual library is then displayed in the Virtual Libraries window. If a particular library still has drives unassigned after this step, the available resources will be displayed in an "Available Library" window. The entire physical library that is being worked with may be configured into virtual libraries. All drives and slots for a selected physical library may be used.

The present system allows the slots in a library to be separated into groups. Grouping slots allows several types of jobs to be run at the same time. In addition, if several slots are available in a group, you can let the library span the media in the group for you. By default, the first time the present system is started, all of the slots in each library that are attached to the machine are automatically assigned to that library group. For example, if there are two libraries attached to the machine, there will be two library groups. All of the slots in the first library will be assigned to GROUP0, and all of the slots in the second library will be assigned to GROUP1. These library group names can be kept, or the user can regroup and rename them. Since each slot in a library is viewed as a virtual storage drive, each slot can be assigned its own group.

The present system allows a user to configure libraries using a GUI. The user can configure (or modify the configuration of) a library by starting a Device Manager. After the system retrieves library device information, the user can select a Device view and locate and select the library device that they wish to configure. The user will be presented with a screen such as that shown in Fig. 21. To configure a device, when a Device Group Configuration window appears, the user clicks the Device tab 161. On the devices window (Fig. 22) there are two sub- windows. On the left side window 160 are the groups and on the right side window 162 are the available devices. the user can take the drive from the Groups list and place it into the Available Devices list. For example, to move the drive from the Group list to the Available Devices list, the user clicks the device they desire to move and clicks Remove button 164. This makes the drive available to any library that the user wants to configure it in. The user then clicks the Library tab 166 and the tape drive you moved is then in the previous tab in the Available Devices list and the changer in the Groups list.

To create a New Group, the user clicks the changer in the list of Groups and clicks New button 168. A New Group dialog appears. The user then types a name for this group that is eight characters or less, chooses a starting and ending slot range, and clicks OK. Groups can also be assigned in a similar manner using Assign button 170.

The present system and method is capable of working in an SAN environment, in which each server that is connected in SAN maintains the information of the shared storage device in the SAN, decides what it wants to do with the media and the tape drive of the device, and lets other servers know about its decision and also shares any change of state in the device initiated by it.

As described above with respect to Fig. 6, one server in the loop is designated as primary and remaining servers are designated as distributed. The primary server 13 is responsible for initializing and inventorying the devices connected over the SAN. The primary server 13 also runs a global locking process called "sanghost" to synchronize access to the storage devices between different servers. The distributed servers 15a-15n take the device information details from the primary server 13 and initialize their internal structures. Once all the servers are up and running they are independent and intelligent enough to make their own decisions. Communication between servers is through a peer system layer that is active on all servers in the SAN.

If one server (e.g., distributed server 15b) wants to back up data and decides to use one drive and media in backup storage device 16a, then it first requests a global lock from the "snapshot" process running on the primary server 13. If no other server is holding this lock, this requesting server gets the lock and can safely go ahead with its operation. Otherwise it has to wait till it can get the global lock. Once server 15b is able to get the global lock, it checks its internal tables, to determine if the drive and media have already been locked by any other server or by itself. If they have not been locked, then server 15b marks that drive and media as locked and propagates that information to other servers on the SAN through its peer system. It then releases the global lock so that other servers can do whatever they want to do with other drives and media. As a server moves media from slot to drive and vice versa, it updates other servers about this action so that they can update their internal tables. In case the media is formatted (or reformatted), the information is propagated to all other servers. After server 15b is finished with its operation, it will again request the global lock, and then tell all other servers to unlock that drive and media. Server 15b will also unlock them from its own internal tables and then release the global lock.

If the primary server 13 crashes, since all other servers know the state of the devices, any other server can be promoted as the primary server on the fly.

The present disclosure may be conveniently implemented using one or more conventional general purpose digital computers and/or servers programmed according to the teachings of the present specification. Appropriate software coding can readily be prepared by skilled programmers based on the teachings of the present disclosure. The present disclosure may also be implemented by the preparation of application specific integrated circuits or by interconnecting an appropriate network of conventional component circuits.

Numerous additional modifications and variations of the present disclosure are possible in view of the above-teachings. It is therefore to be understood that within the scope of the appended claims, the present disclosure may be practiced other than as specifically described herein.

What is Claimed is:

1. A system for backing up and restoring information comprising:

at least one storage device;

a primary server for granting and denying a global lock to devices requesting access to the at least one storage device; and

at least one secondary server including internal tables for storing information relating to storage in the at least one storage device, the at least one secondary server requesting a global lock from the primary server, wherein after the global lock is granted by the primary server, the at least one secondary server checks its own internal tables to determine whether access to the at least one storage device can be granted.

2. The system as recited in claim 1, wherein if it is determined that access to the at least one storage device can be granted, the at least one secondary server marks as locked, at least a portion of the at least one storage device and stores corresponding information in its own internal tables and propagates the corresponding information to the primary server and any other secondary servers for storage in their own internal tables, indicating that the portion of the at least one storage device is locked.

3. The system as recited in claim 2, wherein after the at least one secondary server propagates the corresponding information to the any other secondary servers, it releases the global lock so that the global lock is available to any of the secondary servers or the primary

server.

4. The system as recited in claim 3, wherein after the at least one secondary server releases the global lock, the at least one secondary server begins backing up information into the at least a portion of the at least one storage device or restoring information from the at least a portion of the at least one storage device.

5. The system as recited in claim 4, wherein after the at least one secondary server is done backing up into or restoring information from the at least a portion of the at least one storage device, the at least one secondary server again requests the global lock so that it can update the primary server and any other secondary servers accordingly.

6. The system as recited in claim 5, wherein after the at least one secondary server is again given the global lock, the at least one secondary server sends information to the primary server and any other secondary servers informing them to update their internal tables to unlock the at least a portion of the at least one storage device.

7. The system as recited in claim 1, further comprising a network for connecting the at least one storage device, the primary server and the plurality of secondary servers.

8. The system as recited in claim 7, wherein the network comprises a storage area network.

9. A method of granting and denying access to servers on a network for backing up information to and restoring information from a storage device, the servers including a primary server and secondary servers, comprising:

sending from a secondary server to the primary server a request for a global lock, the secondary server including internal tables for storing information relating to storage in the storage device;

determining whether any other servers on the network have the global lock;

granting and denying the global lock to the secondary server based on whether any other servers on the network have the global lock,

wherein when the global lock is granted by the primary server, the at least one secondary server checking its own internal tables to determine whether access to the at least one storage device can be granted.

10. The system as recited in claim 9, wherein if it is determined that access to the at least one storage device can be granted, the at least one secondary server marks as locked, at least a portion of the at least one storage device and stores corresponding information in its own internal tables and propagates the corresponding information to the primary server and any other secondary servers for storage in their own internal tables, indicating that the portion of the at least one storage device is locked.

11. The system as recited in claim 10, wherein after the at least one secondary server propagates the corresponding information to the any other secondary servers, it releases the global lock so that the global lock is available to any of the secondary servers or the primary server.

12. The system as recited in claim 11, wherein after the at least one secondary server releases the global lock, the at least one secondary server begins backing up information into the at least a portion of the at least one storage device or restoring information from the at least a portion of the at least one storage device.

13. The system as recited in claim 12, wherein after the at least one secondary server is done backing up into or restoring information from the at least a portion of the at least one storage device, the at least one secondary server again requests the global lock so that it can update the primary server and any other secondary servers accordingly.

14. The system as recited in claim 14, wherein after the at least one secondary server is again given the global lock, the at least one secondary server sends information to the primary server and any other secondary servers informing them to update their internal tables to unlock the at least a portion of the at least one storage device.

15. The system as recited in claim 9, wherein the network comprises a storage area network.

16. A computer recording medium including computer executable code for granting and denying access to servers on a network for backing up information to and restoring information from a storage device, the servers including a primary server and secondary servers, comprising:

code for sending from a secondary server to the primary server a request for a global lock, the secondary server including internal tables for storing information relating to storage in the storage device;

code for determining whether any other servers on the network have the global lock;

code for granting and denying the global lock to the secondary server based on whether any other servers on the network have the global lock,

wherein when the global lock is granted by the primary server, the at least one secondary server checking its own internal tables to determine whether access to the at least one storage device can be granted.

17. The computer recording medium as recited in claim 16, further comprising code so that if it is determined that access to the at least one storage device can be granted, the at least one secondary server is controlled to mark as locked, at least a portion of the at least one storage device and store corresponding information in its own internal tables and code for propagating the corresponding information to the primary server and any other secondary servers for storage in their own internal tables, indicating that the portion of the at least one storage device is locked.

18. The computer recording medium as recited in claim 17, wherein after the at least one secondary server propagates the corresponding information to the any other secondary servers, it

releases the global lock so that the global lock is available to any of the secondary servers or the primary server.

19. The computer recording medium as recited in claim 18, further comprising code for controlling the at least one secondary server, so that after the at least one secondary server releases the global lock, the at least one secondary server begins backing up information into the at least a portion of the at least one storage device or restoring information from the at least a portion of the at least one storage device.

20. The computer recording medium as recited in claim 19, further comprising code for controlling the at least one secondary server, so that after the at least one secondary server is done backing up into or restoring information from the at least a portion of the at least one storage device, the at least one secondary server again requests the global lock so that it can update the primary server and any other secondary servers accordingly.

21. The computer recording medium as recited in claim 20, further comprising code for controlling the at least one secondary server, so that after the at least one secondary server is again given the global lock, the at least one secondary server sends information to the primary server and any other secondary servers informing them to update their internal tables to unlock the at least a portion of the at least one storage device.

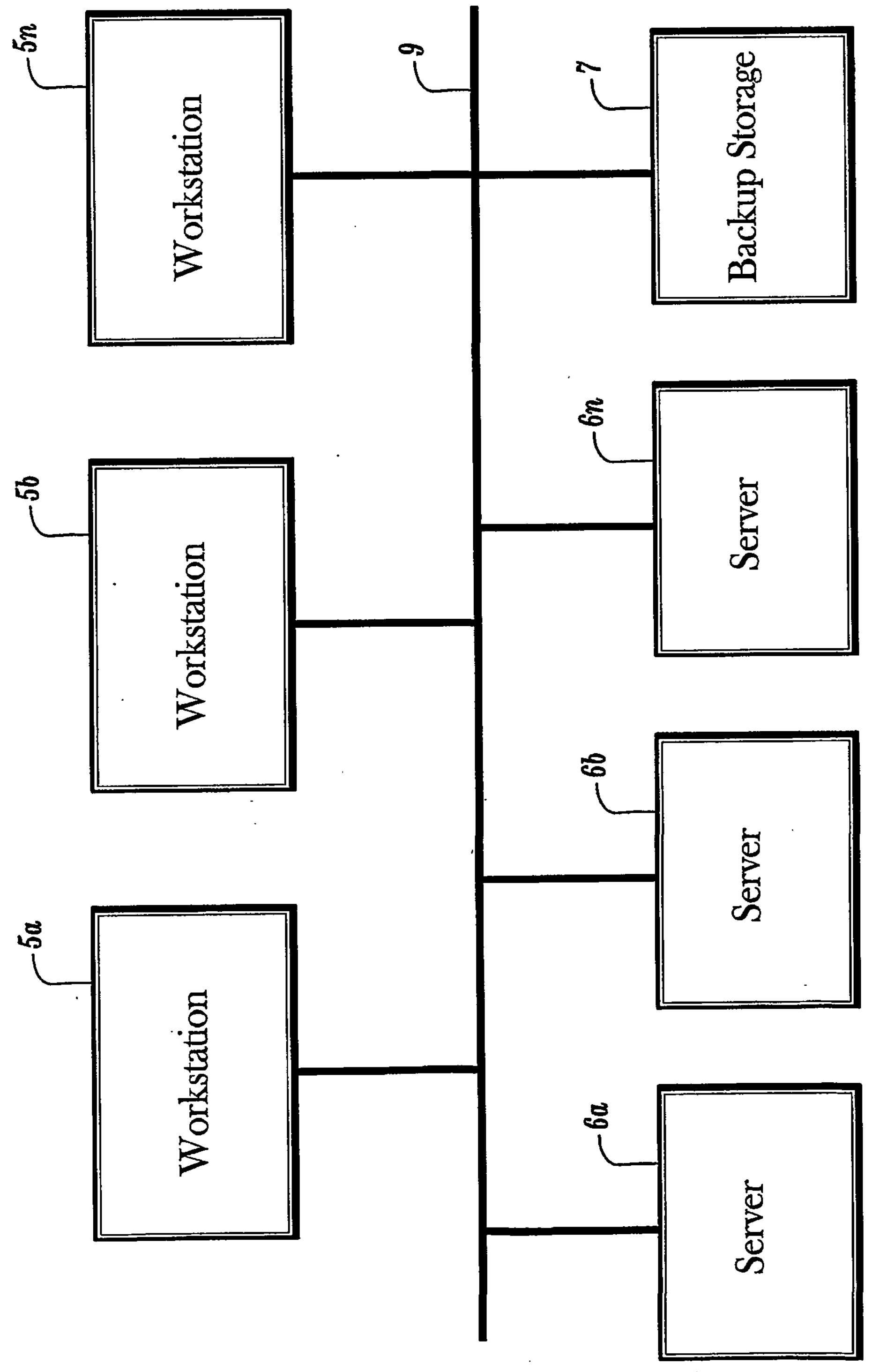

5n
Workstation
5b
Workstation
5a
Workstation
9
7
Backup Storage
6n
Server
6b
Server
6a
Server

FIG. 1A

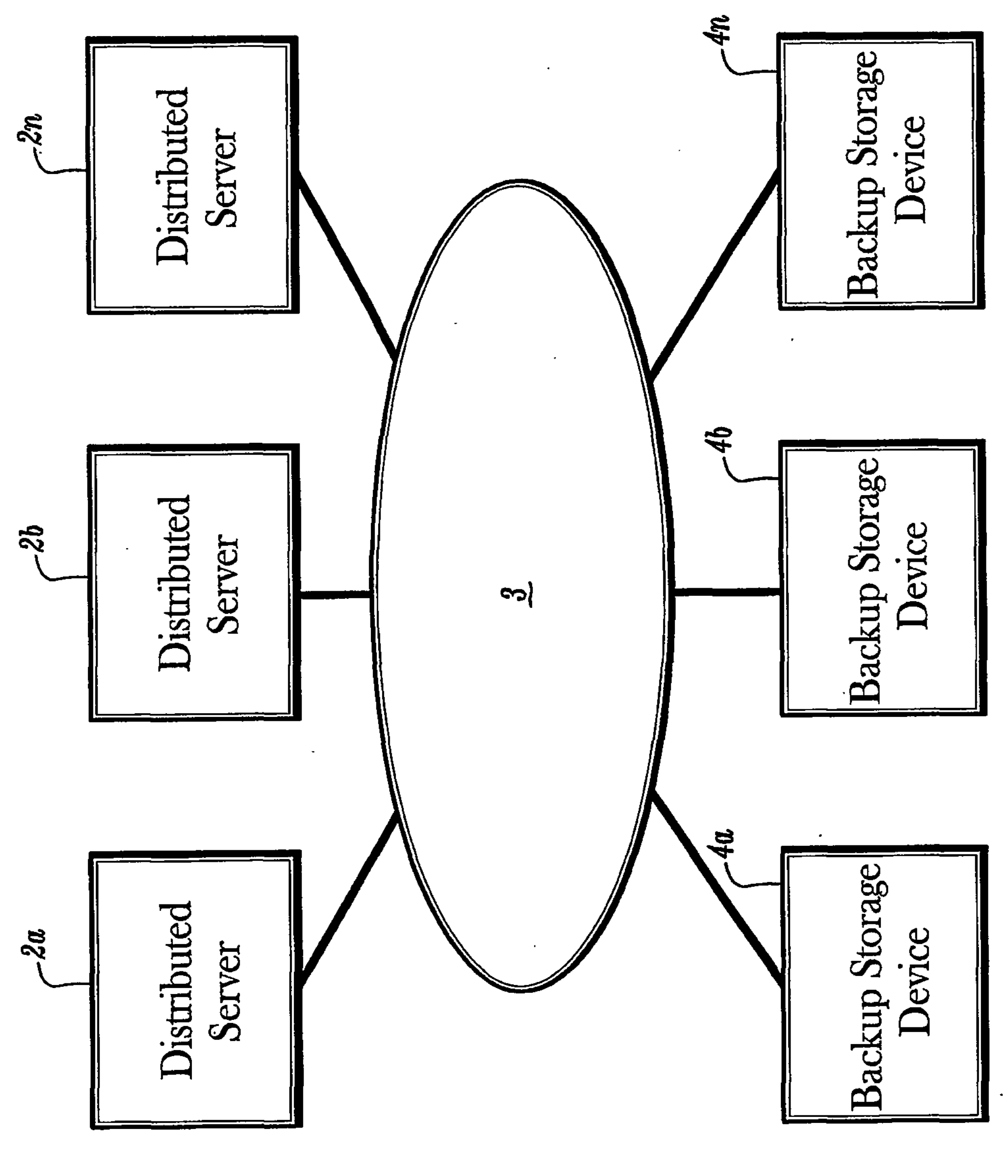

2a
Distributed Server
2b
Distributed Server
2n
Distributed Server
3
4a
Backup Storage Device
4b
Backup Storage Device
4n
Backup Storage Device

FIG. 1B

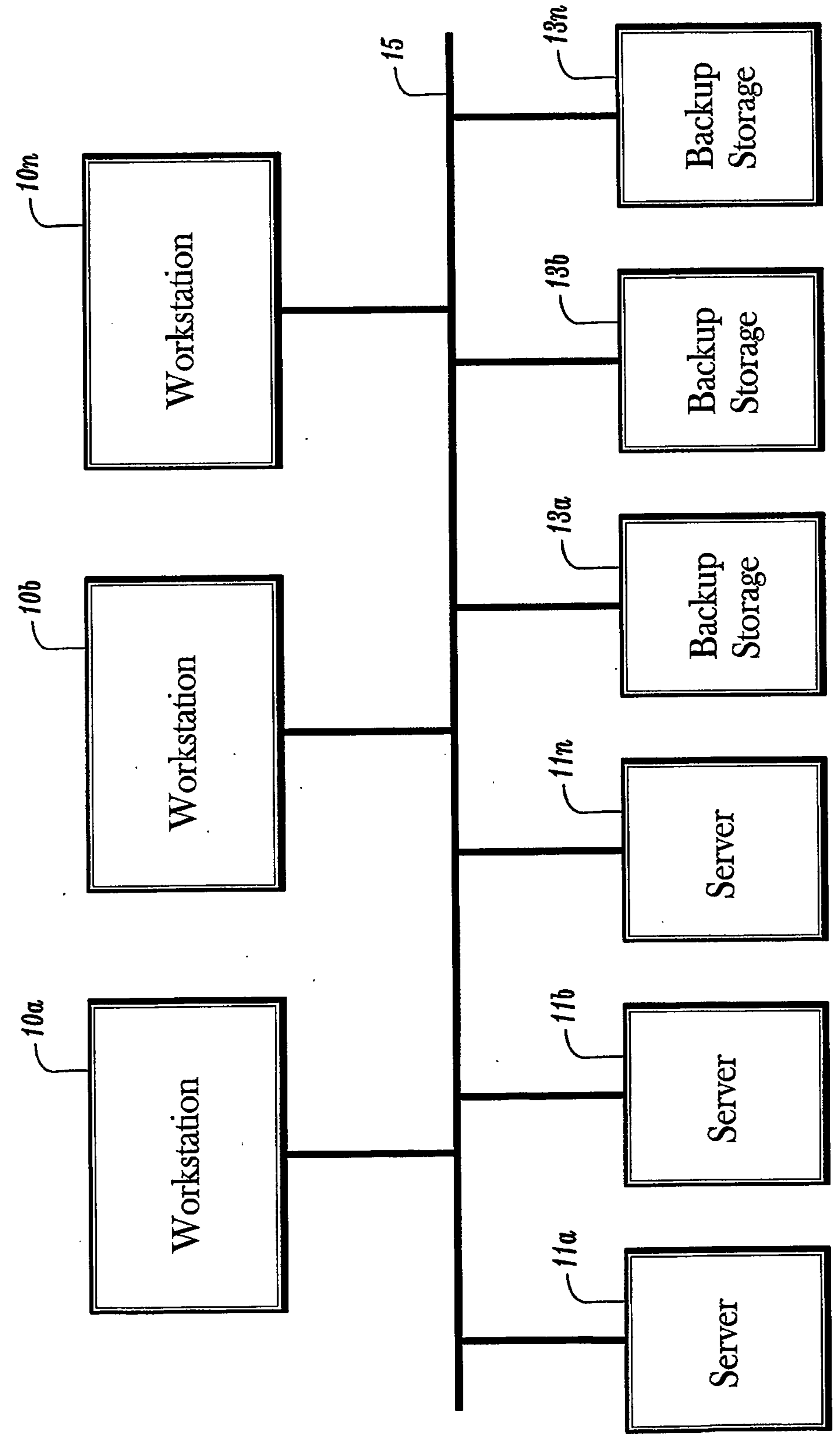

15
10a
Workstation
10b
Workstation
10n
Workstation
11a
Server
11b
Server
11n
Server
13a
Backup Storage
13b
Backup Storage
13n
Backup Storage

FIG. 2

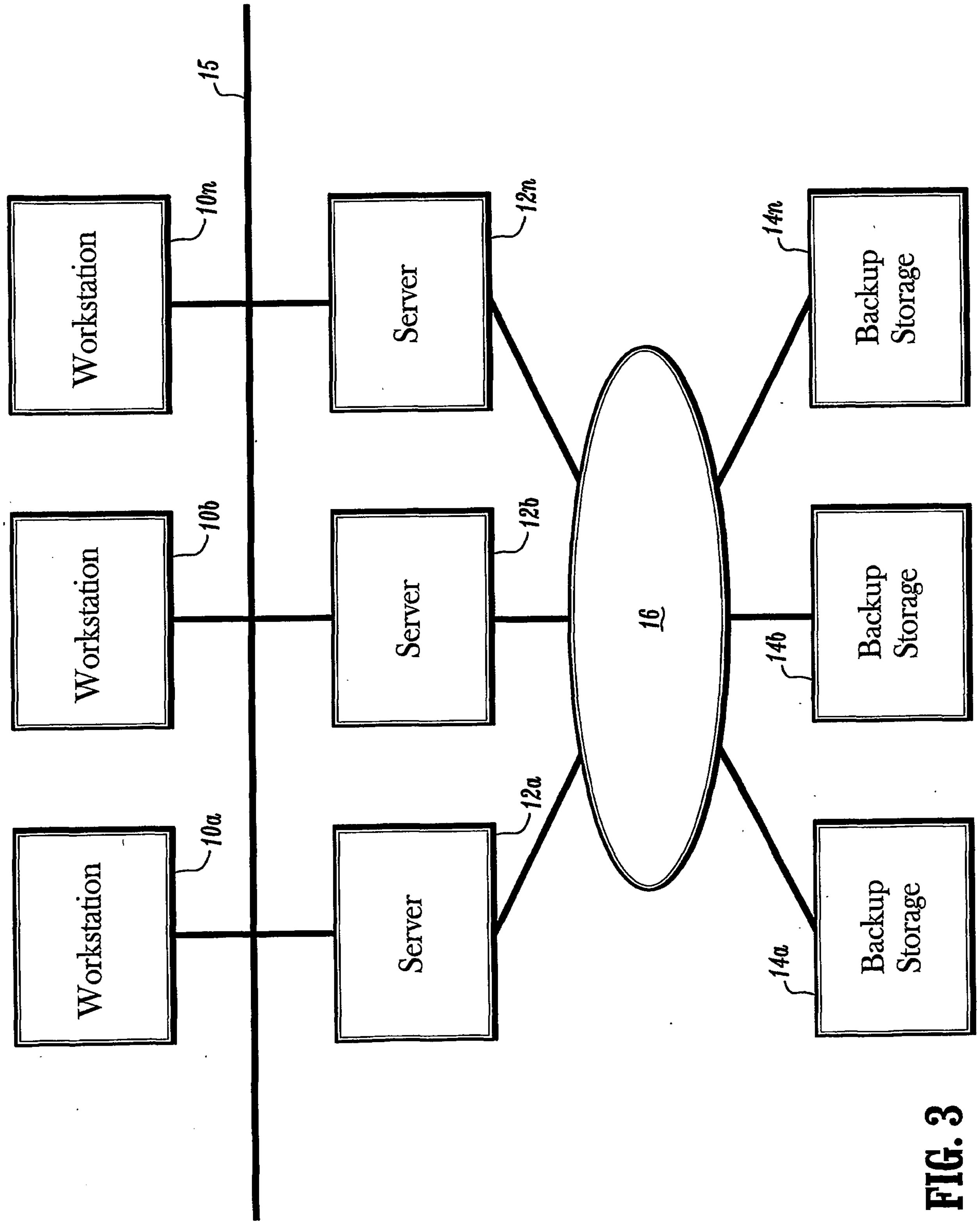

15
Workstation
10n
Workstation
10b
Workstation
10a
Server
12n
Server
12b
Server
12a
16
Backup Storage
14n
Backup Storage
14b
Backup Storage
14a

FIG. 3

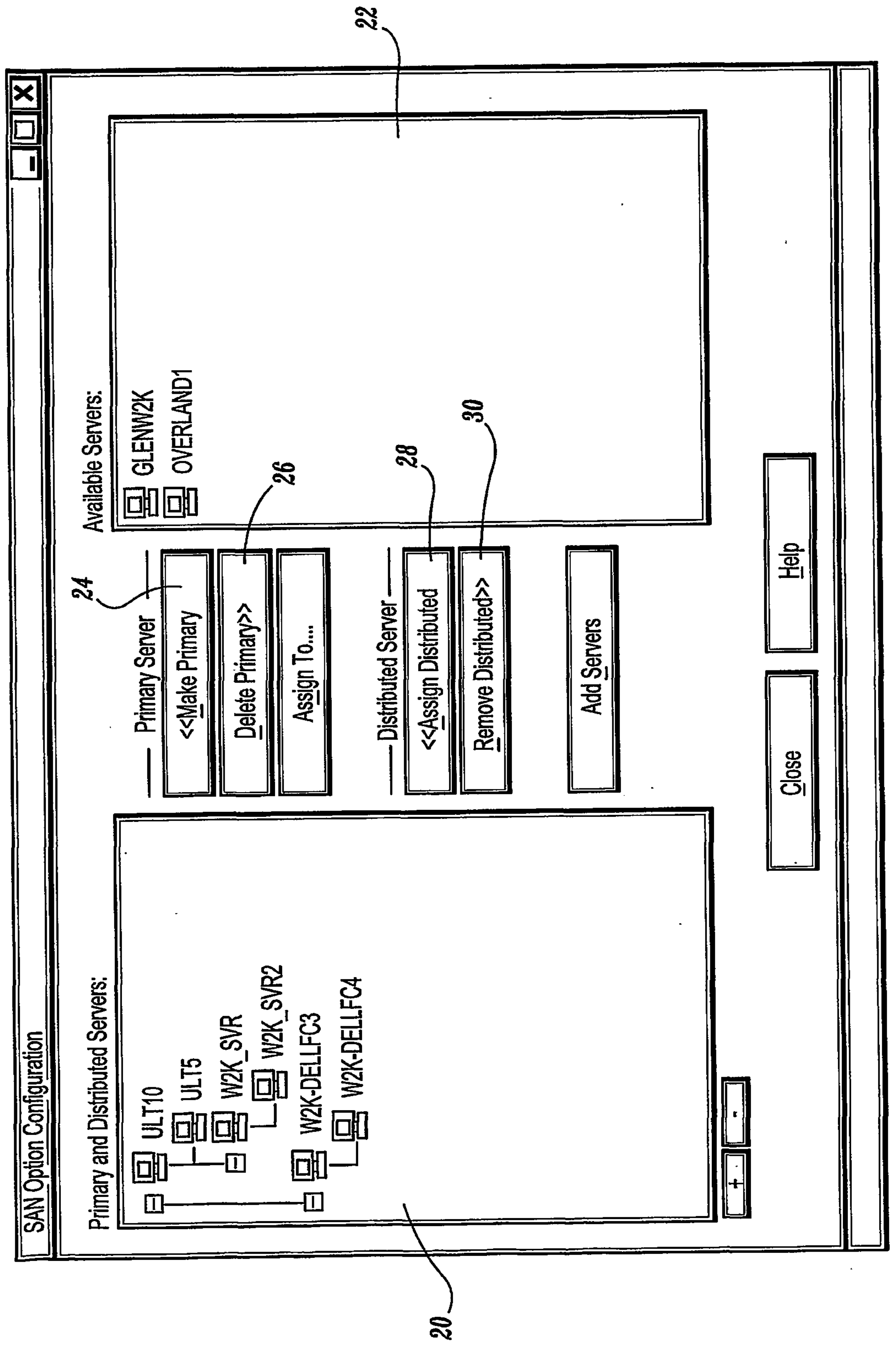
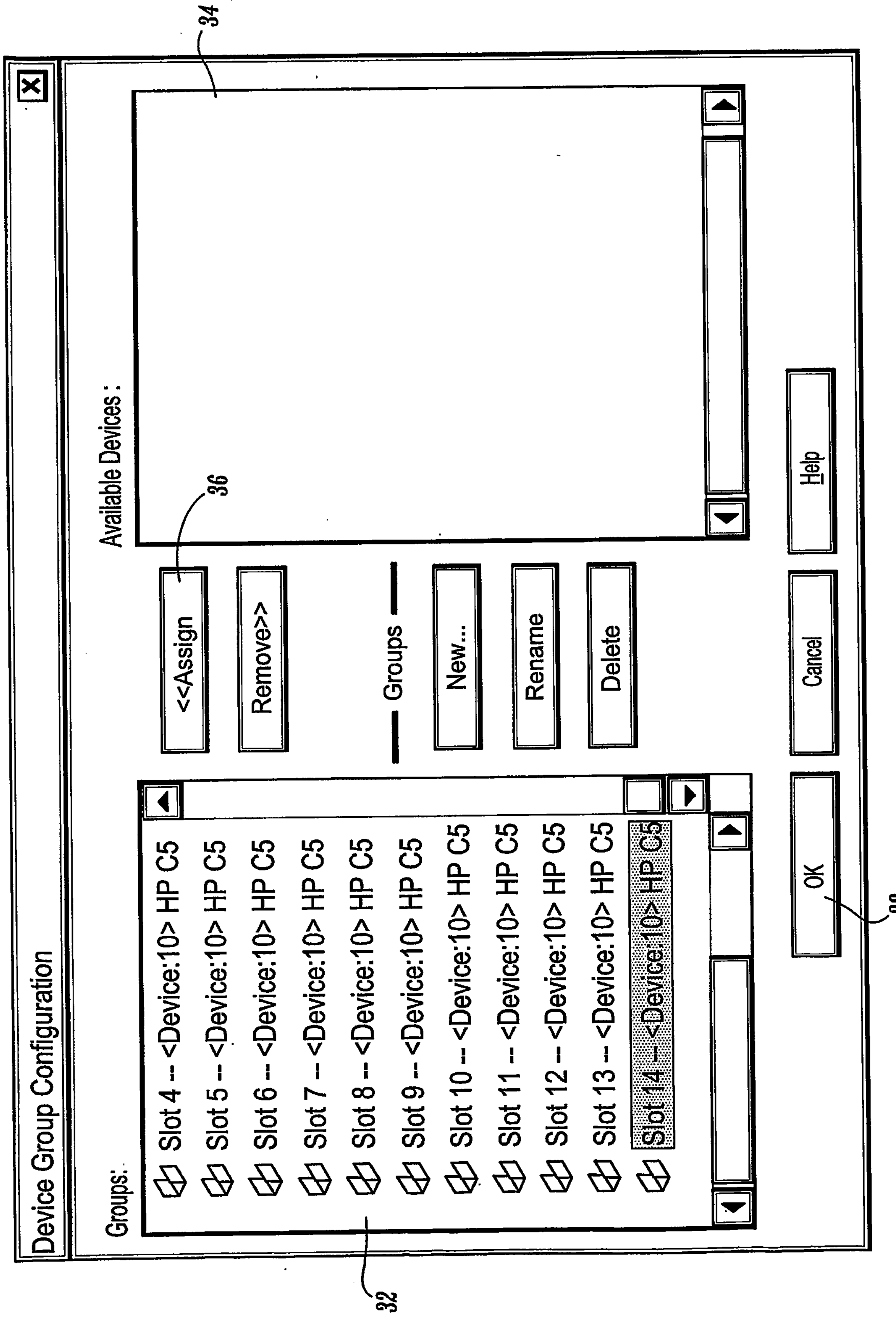

SAN Option Configuration
Primary and Distributed Servers:
ULT10
ULT5
W2K_SVR
W2K_SVR2
W2K-DELLFC3
W2K-DELLFC4
+
-
Primary Server
<<Make Primary
Delete Primary>>
Assign To....
Distributed Server
<<Assign Distributed
Remove Distributed>>
Add Servers
Available Servers:
GLENW2K
OVERLAND1
Close
Help
20
22
24
26
28
30

FIG. 4



Slot 4 -- <Device:10> HP C5
Slot 5 -- <Device:10> HP C5
Slot 6 -- <Device:10> HP C5
Slot 7 -- <Device:10> HP C5
Slot 8 -- <Device:10> HP C5
Slot 9 -- <Device:10> HP C5
Slot 10 -- <Device:10> HP C5
Slot 11 -- <Device:10> HP C5
Slot 12 -- <Device:10> HP C5
Slot 13 -- <Device:10> HP C5
Slot 14 -- <Device:10> HP C5

FIG. 5

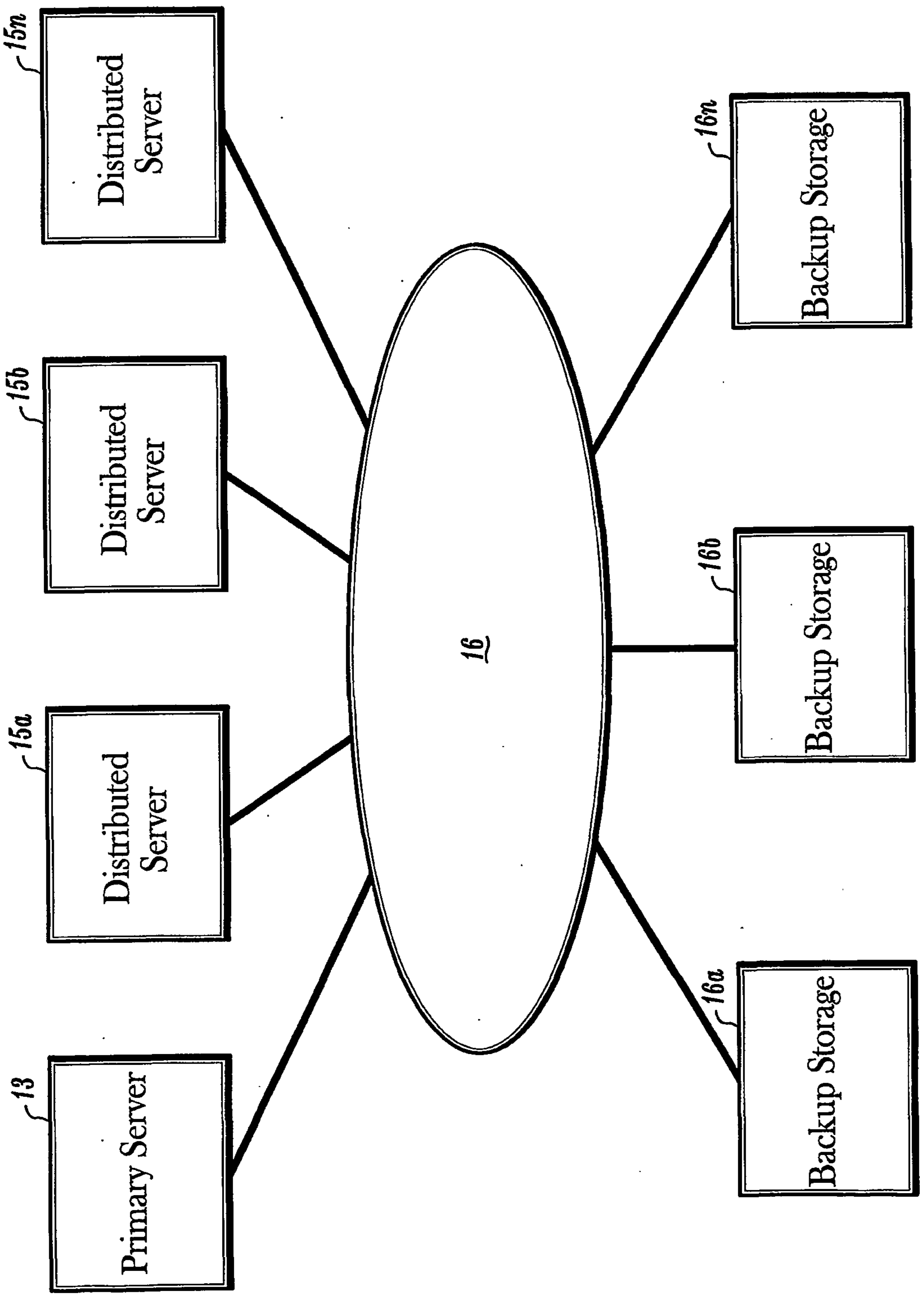
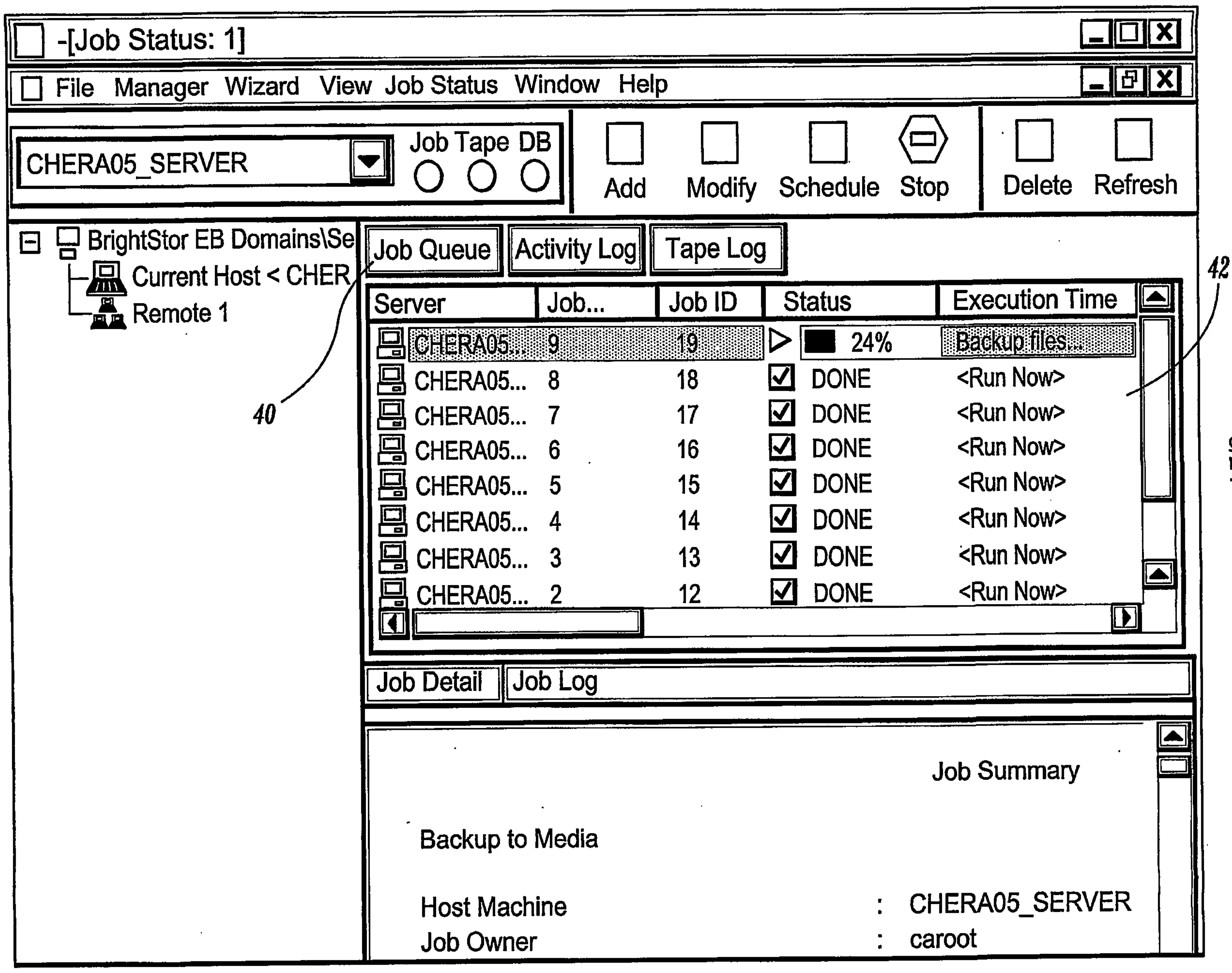

13
Primary Server
15a
Distributed Server
15b
Distributed Server
15n
Distributed Server
16
16a
Backup Storage
16b
Backup Storage
16n
Backup Storage

FIG. 6



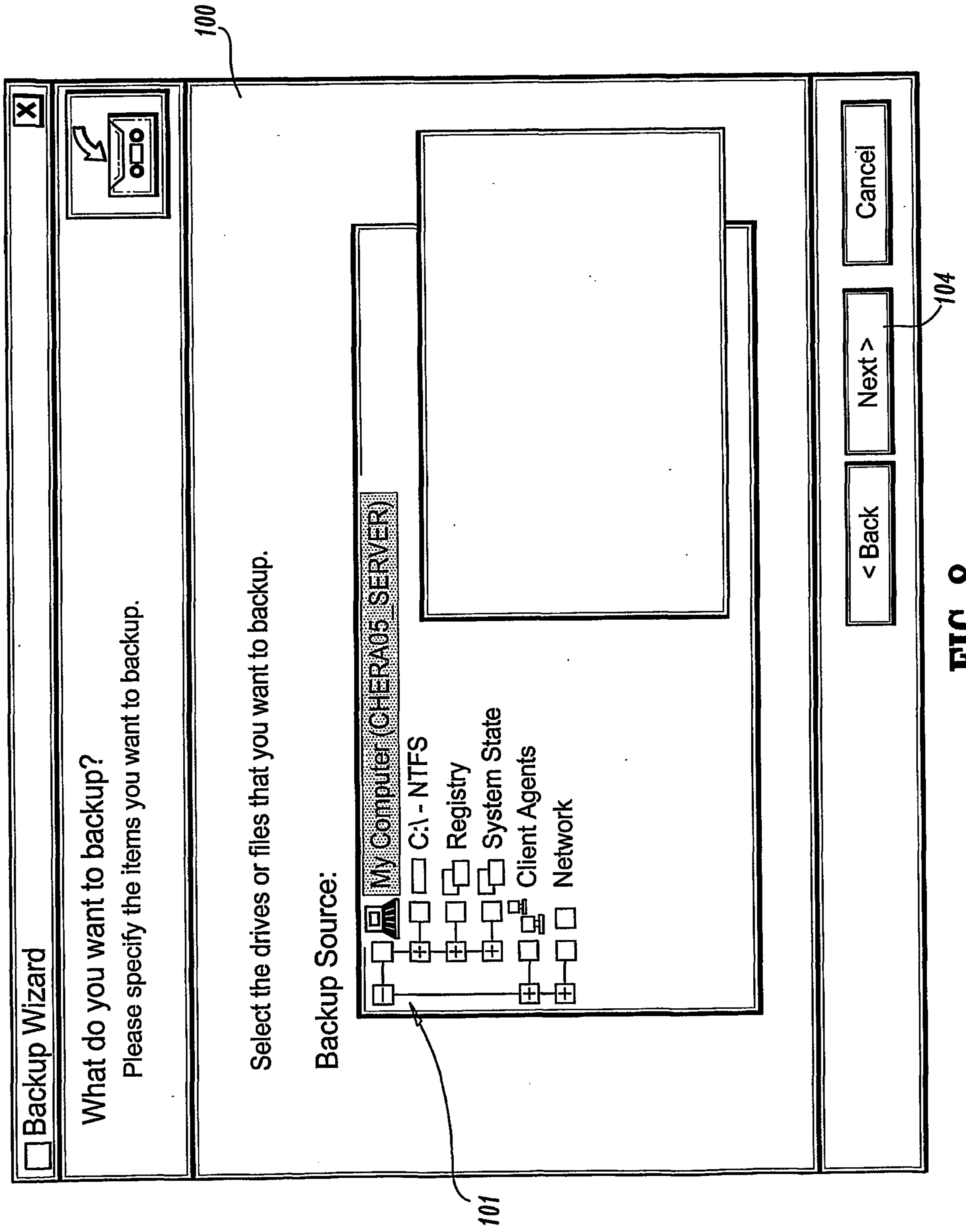
Server	Job...	Job ID	Status	Execution Time
CHERA05...	9	19	▷ ■ 24%	Backup files...
CHERA05...	8	18	✓ DONE	<Run Now>
CHERA05...	7	17	✓ DONE	<Run Now>
CHERA05...	6	16	✓ DONE	<Run Now>
CHERA05...	5	15	✓ DONE	<Run Now>
CHERA05...	4	14	✓ DONE	<Run Now>
CHERA05...	3	13	✓ DONE	<Run Now>
CHERA05...	2	12	✓ DONE	<Run Now>

Job Summary

Backup to Media

Host Machine	:	CHERA05_SERVER
Job Owner	:	caroot

FIG. 7



Backup Wizard

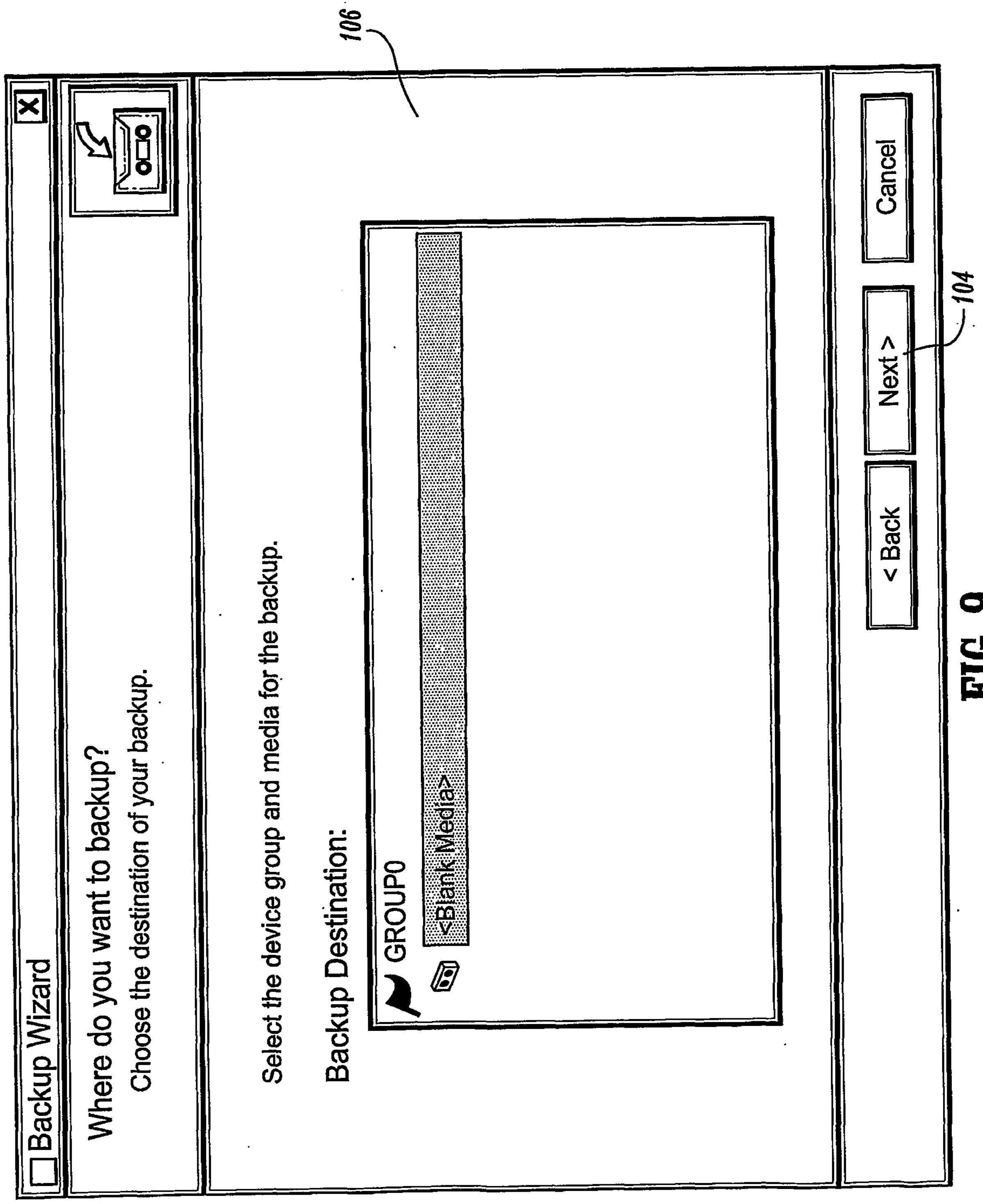
What do you want to backup?

Please specify the items you want to backup.

Select the drives or files that you want to backup.

Backup Source:

FIG. 8



Backup Wizard

Where do you want to backup?

Choose the destination of your backup.

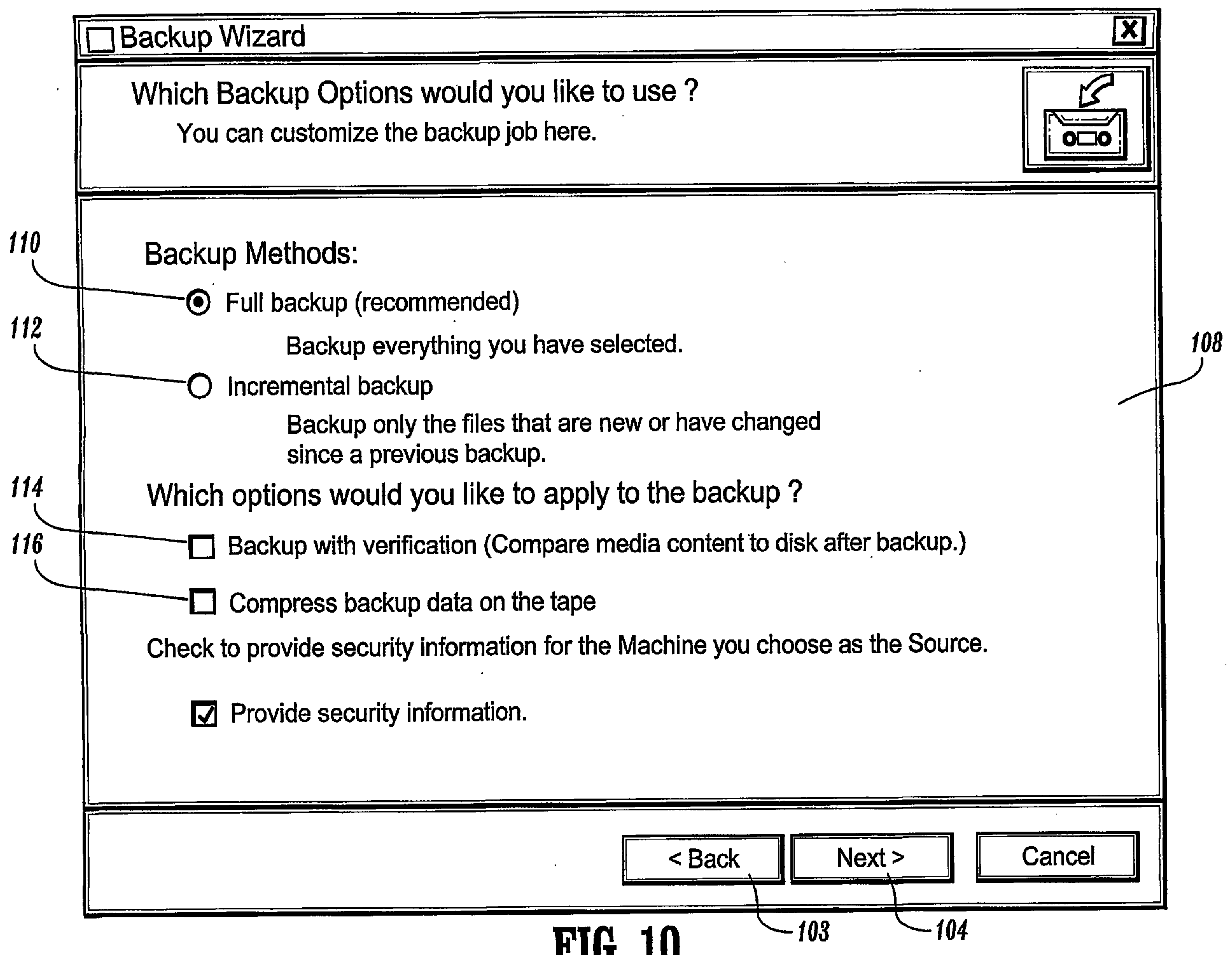
Select the device group and media for the backup.

Backup Destination:

GROUP0

<Blank Media>

FIG. 9



Which Backup Options would you like to use ?
You can customize the backup job here.

Backup Methods:

- [✓ Full backup (recommended)]
  Backup everything you have selected.
- Incremental backup
  Backup only the files that are new or have changed since a previous backup.

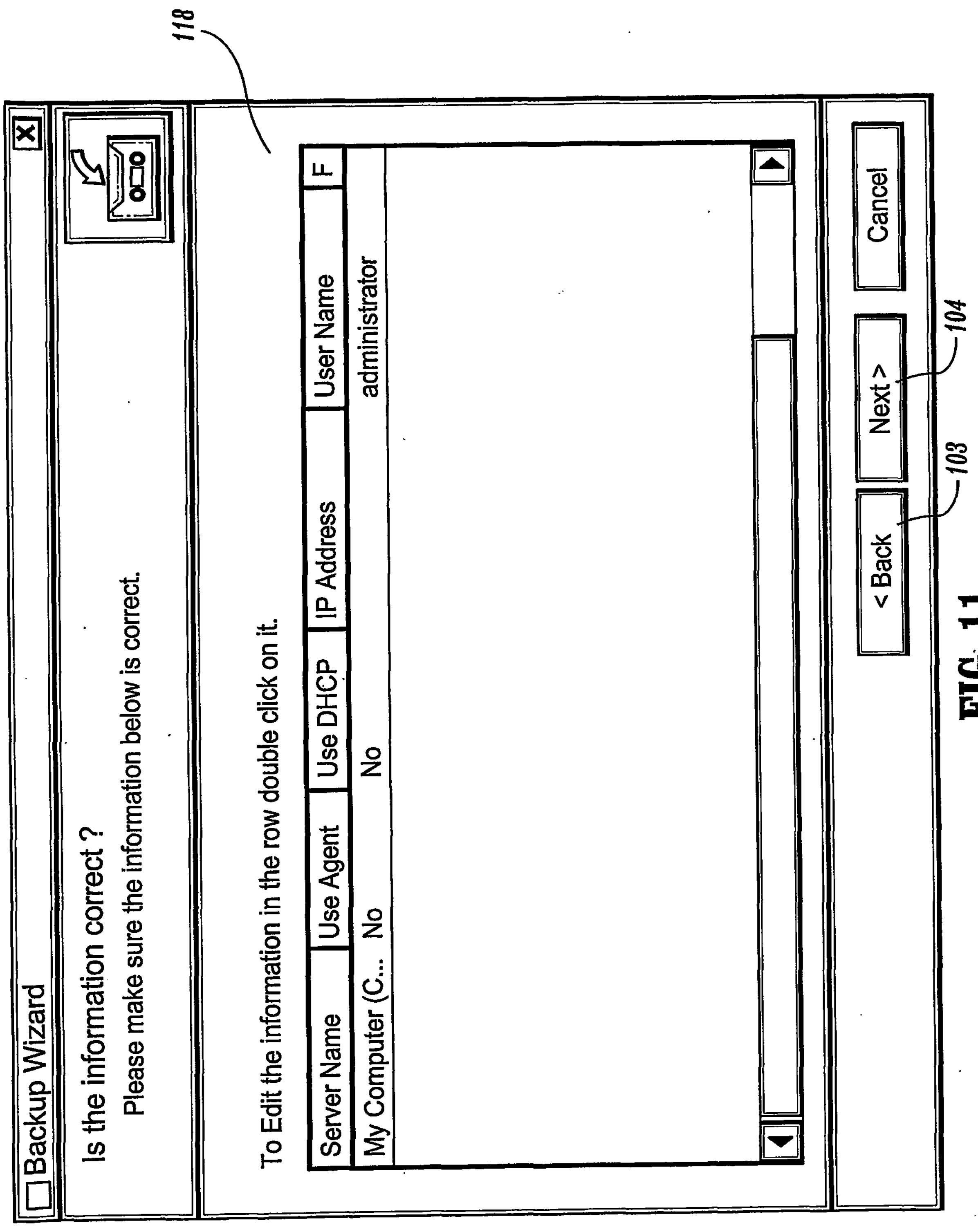
Which options would you like to apply to the backup ?

- ☐ Backup with verification (Compare media content to disk after backup.)
- ☐ Compress backup data on the tape

Check to provide security information for the Machine you choose as the Source.

- [✓ Provide security information.]

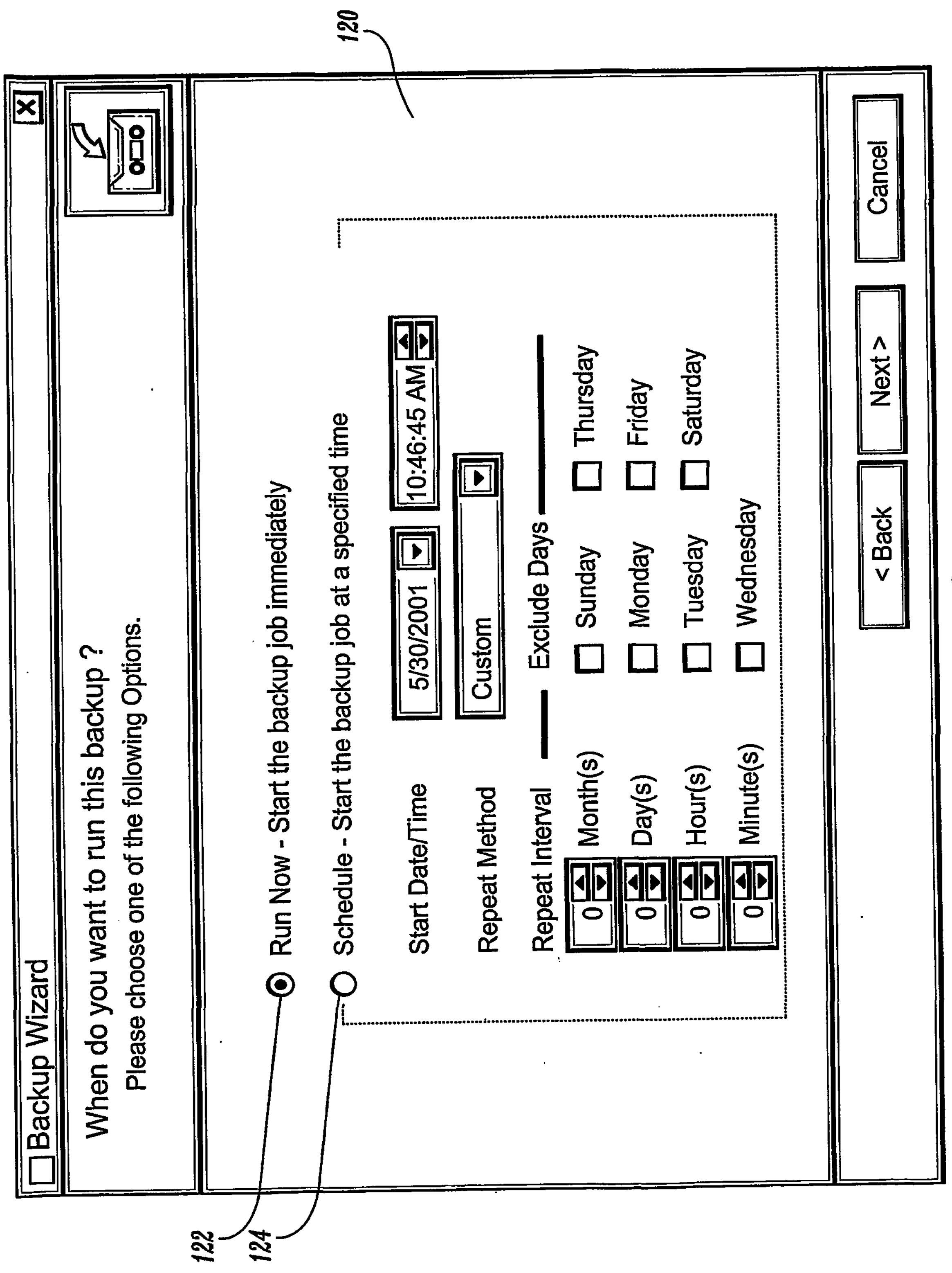
FIG. 10



Backup Wizard
Is the information correct ?
Please make sure the information below is correct.
To Edit the information in the row double click on it.
< Back
Next >
Cancel
118
103
104

Server Name	Use Agent	Use DHCP	IP Address	User Name	F
My Computer (C...	No	No		administrator	

FIG. 11



When do you want to run this backup ?
Please choose one of the following Options.

[✓ Run Now - Start the backup job immediately]

Schedule - Start the backup job at a specified time

Start Date/Time 5/30/2001 10:46:45 AM

Repeat Method Custom

Repeat Interval
0 Month(s)
0 Day(s)
0 Hour(s)
0 Minute(s)

Exclude Days
☐ Sunday ☐ Thursday
☐ Monday ☐ Friday
☐ Tuesday ☐ Saturday
☐ Wednesday

FIG. 12

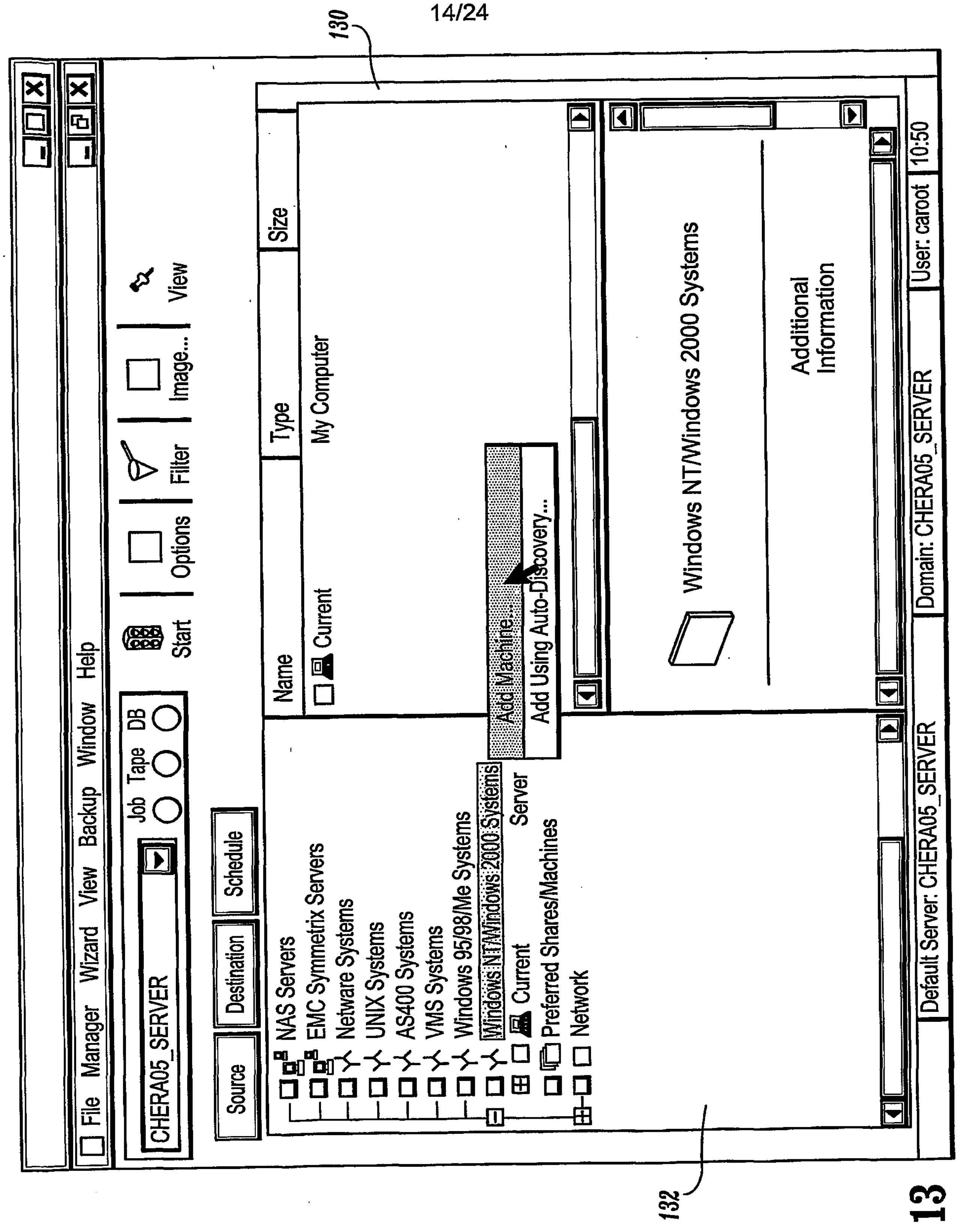

130
132
File Manager Wizard View Backup Window Help
CHERA05_SERVER
Job Tape DB
Start
Options
Filter
Image...
View
Source
Destination
Schedule
NAS Servers
EMC Symmetrix Servers
Netware Systems
UNIX Systems
AS400 Systems
VMS Systems
Windows 95/98/Me Systems
Windows NT/Windows 2000 Systems
Current Server
Preferred Shares/Machines
Network
Name
Type
Size
Current
My Computer
Add Machine...
Add Using Auto-Discovery...
Windows NT/Windows 2000 Systems
Additional Information
Default Server: CHERA05_SERVER
Domain: CHERA05_SERVER
User: caroot
10:50

FIG. 13

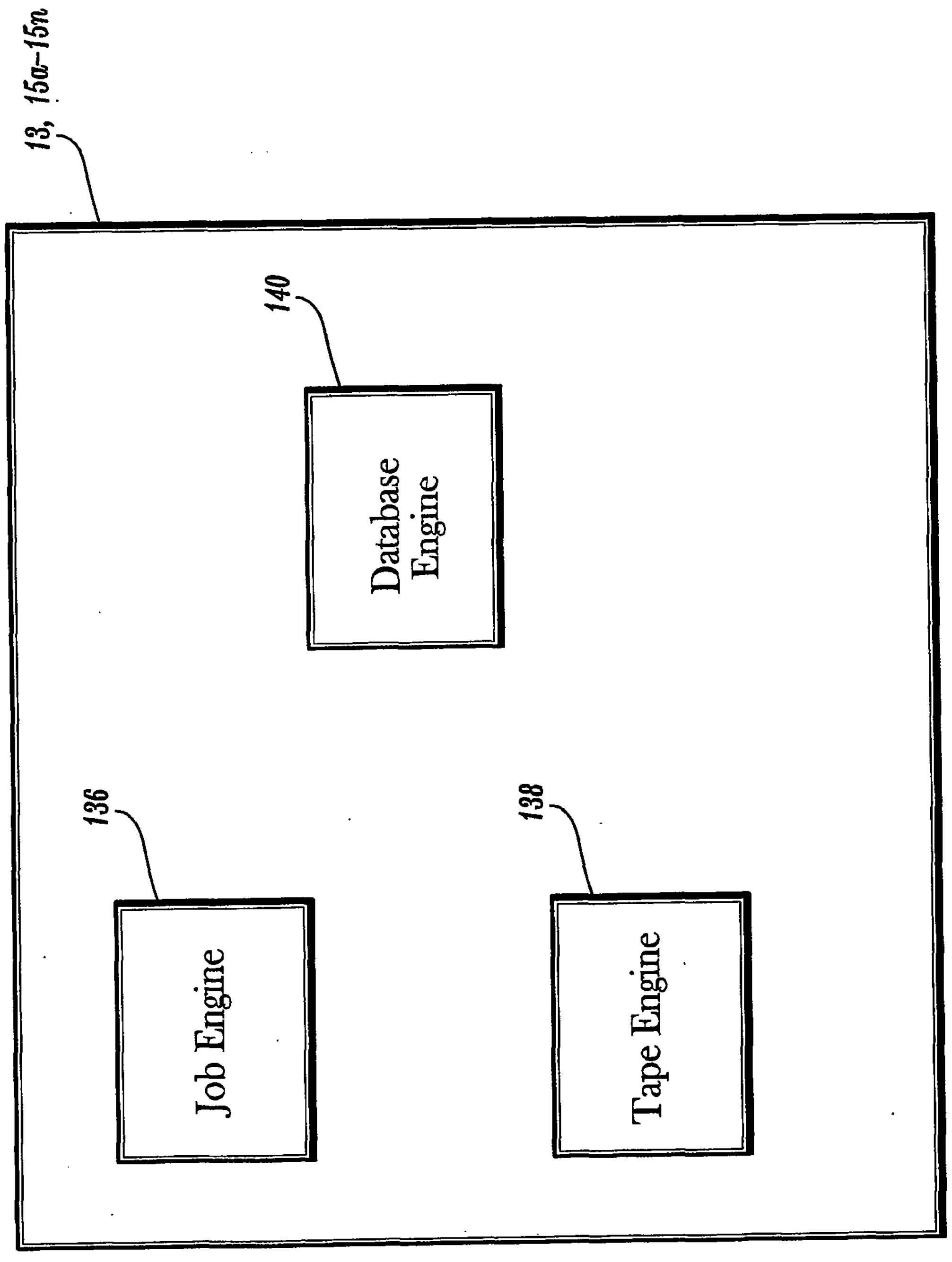

13, 15a–15n
140
Database Engine
136
Job Engine
138
Tape Engine

FIG. 14

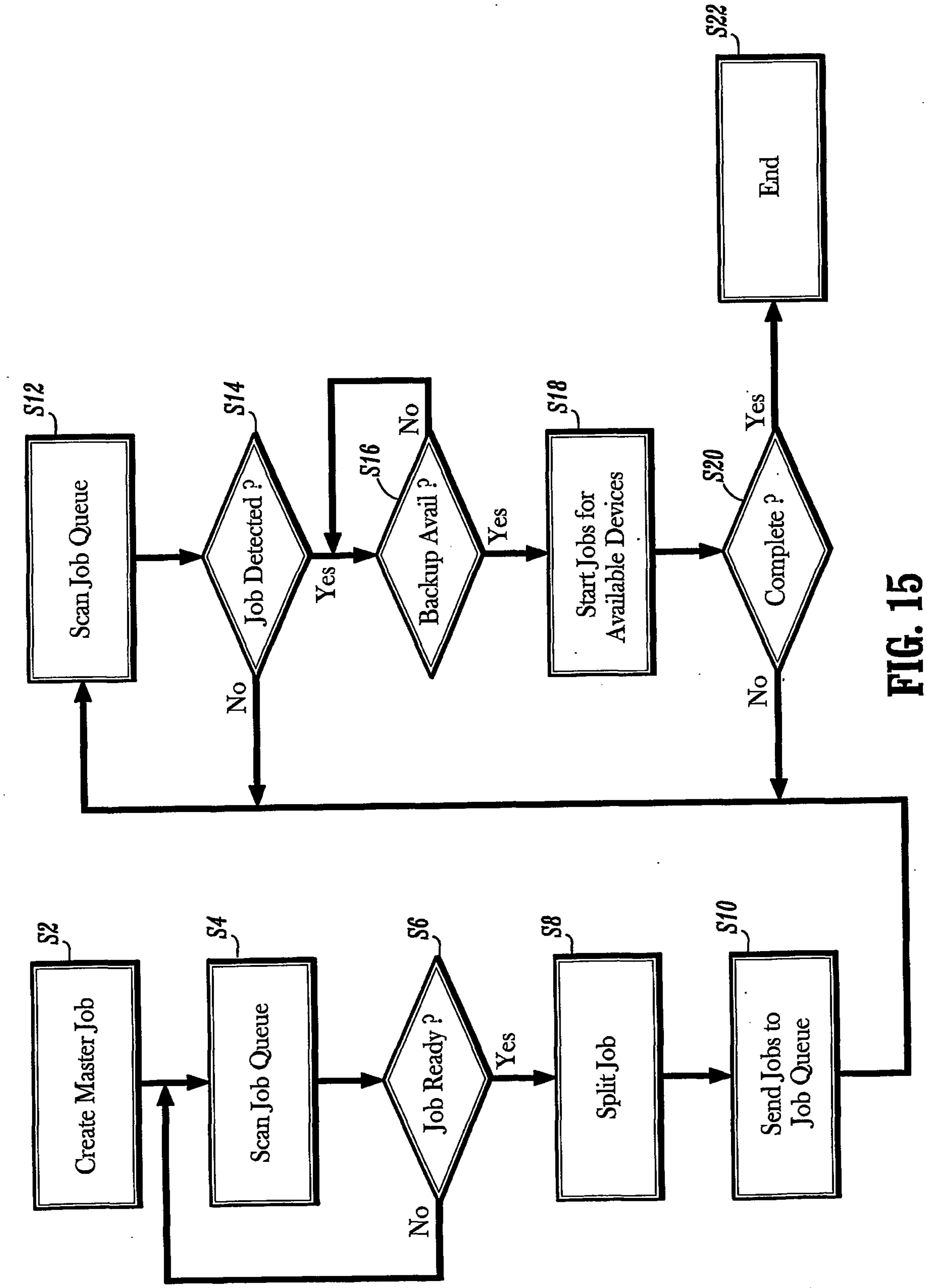

S2
Create Master Job
S4
Scan Job Queue
S6
Job Ready ?
Yes
No
S8
Split Job
S10
Send Jobs to Job Queue
S12
Scan Job Queue
S14
Job Detected ?
Yes
No
S16
Backup Avail ?
Yes
No
S18
Start Jobs for Available Devices
S20
Complete ?
Yes
No
S22
End

FIG. 15

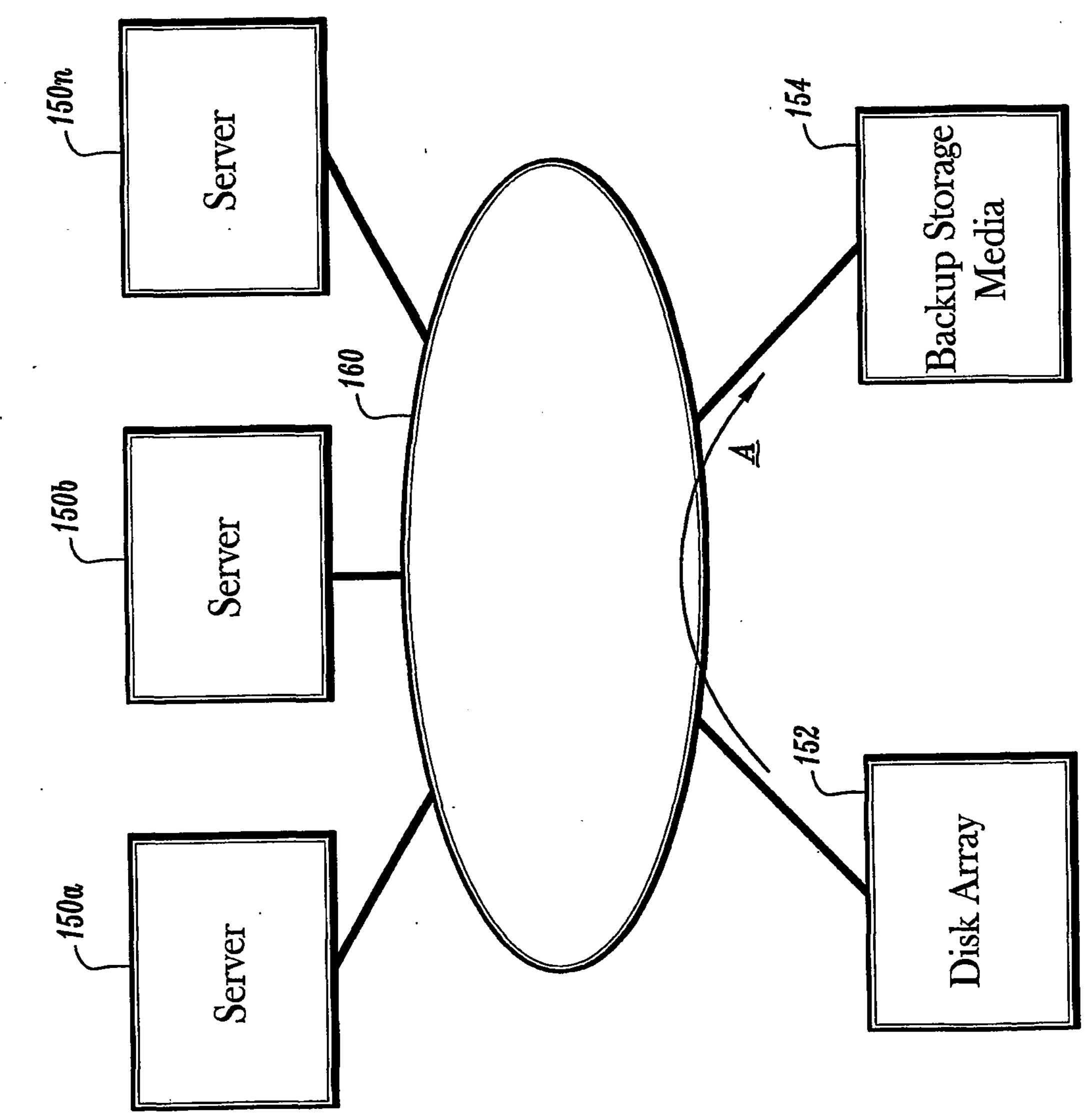

150n
Server
150b
Server
150a
Server
160
154
Backup Storage Media
A
152
Disk Array

FIG. 16

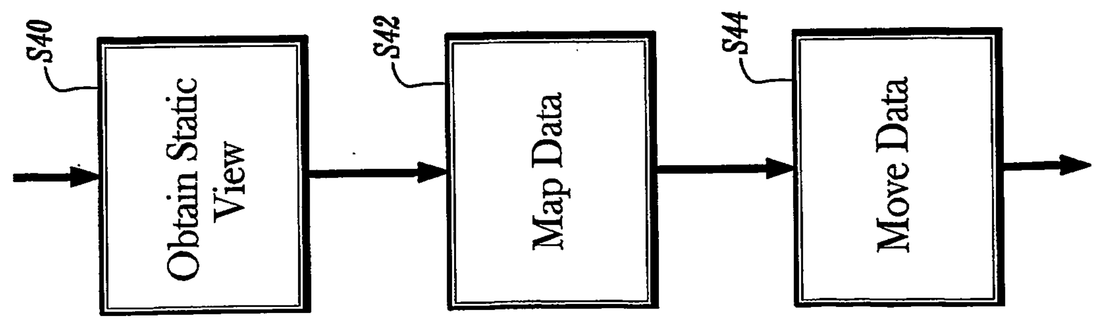


FIG. 17

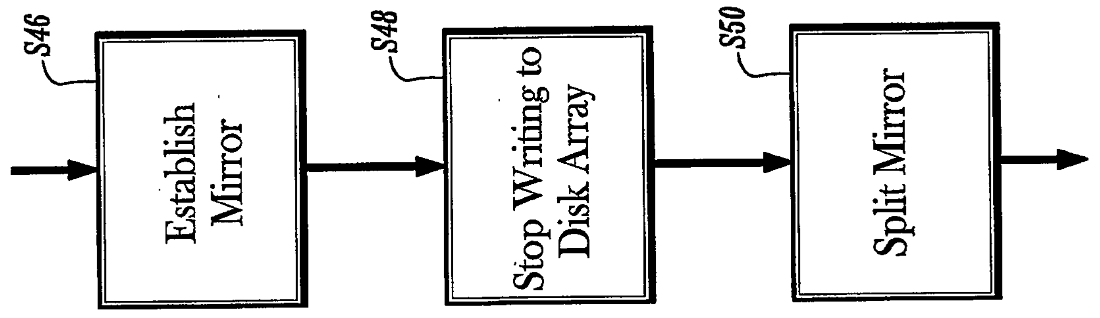


FIG. 18

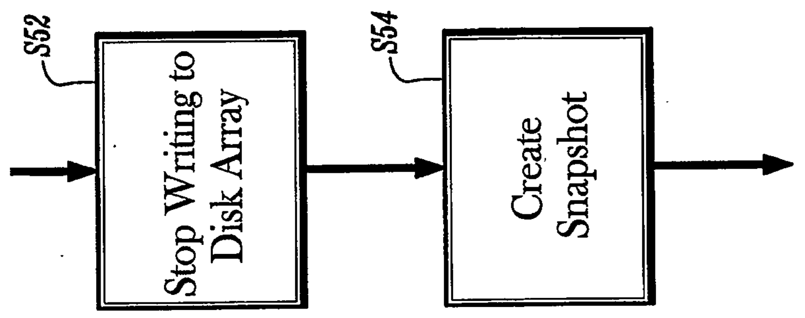


FIG. 19

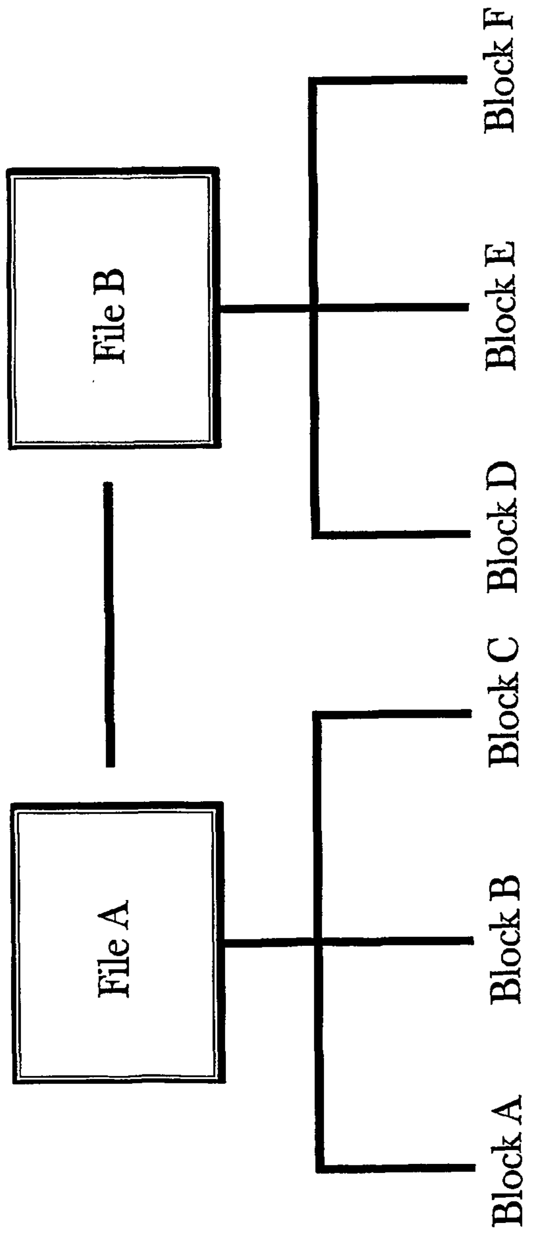
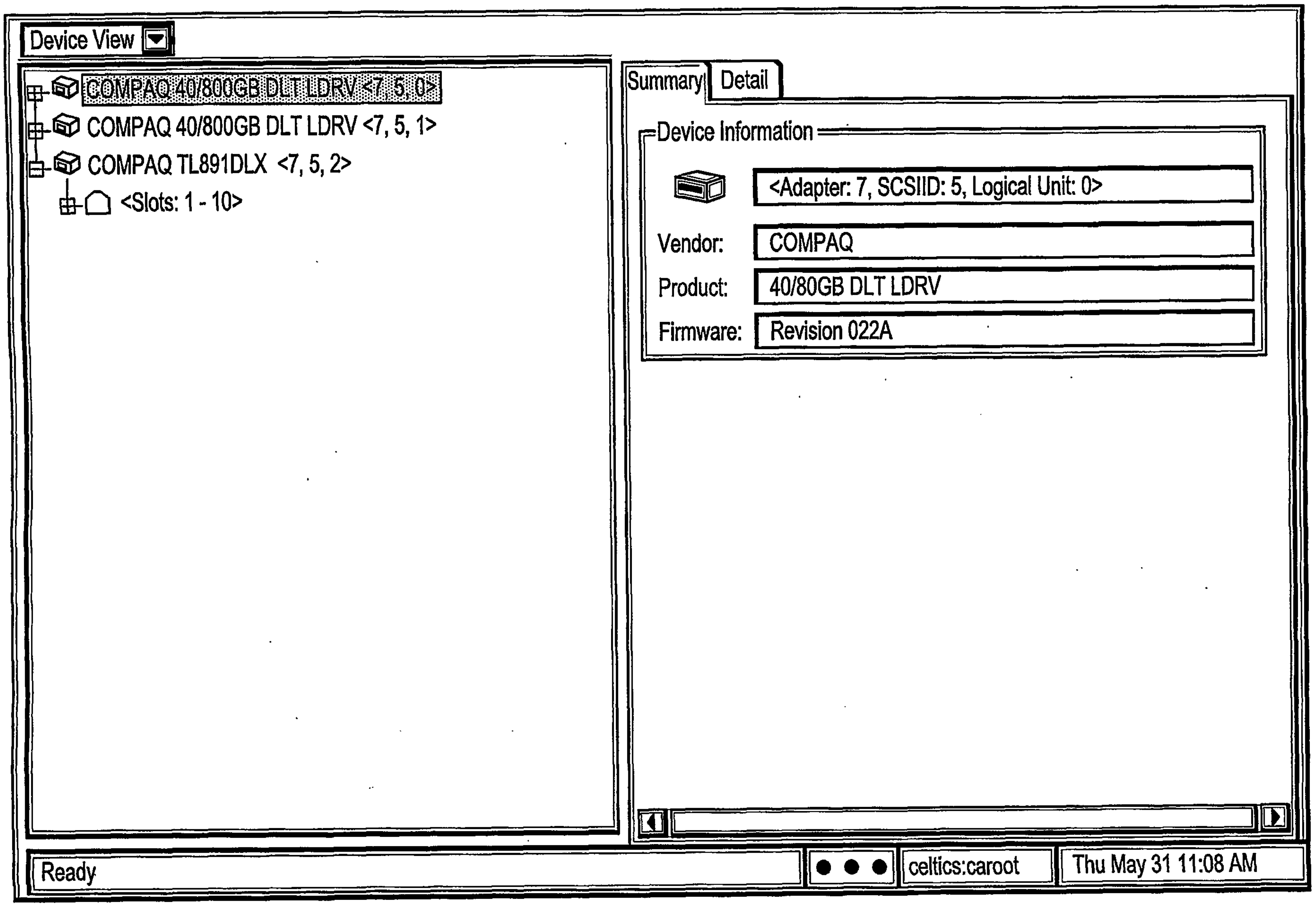


FIG. 20



COMPAQ 40/800GB DLT LDRV <7, 5, 0>
COMPAQ 40/800GB DLT LDRV <7, 5, 1>
COMPAQ TL891DLX <7, 5, 2>
<Slots: 1 - 10>

Device Information

	<Adapter: 7, SCSIID: 5, Logical Unit: 0>
Vendor:	COMPAQ
Product:	40/80GB DLT LDRV
Firmware:	Revision 022A

Ready | ● ● ● | celtics:caroot | Thu May 31 11:08 AM

FIG. 21

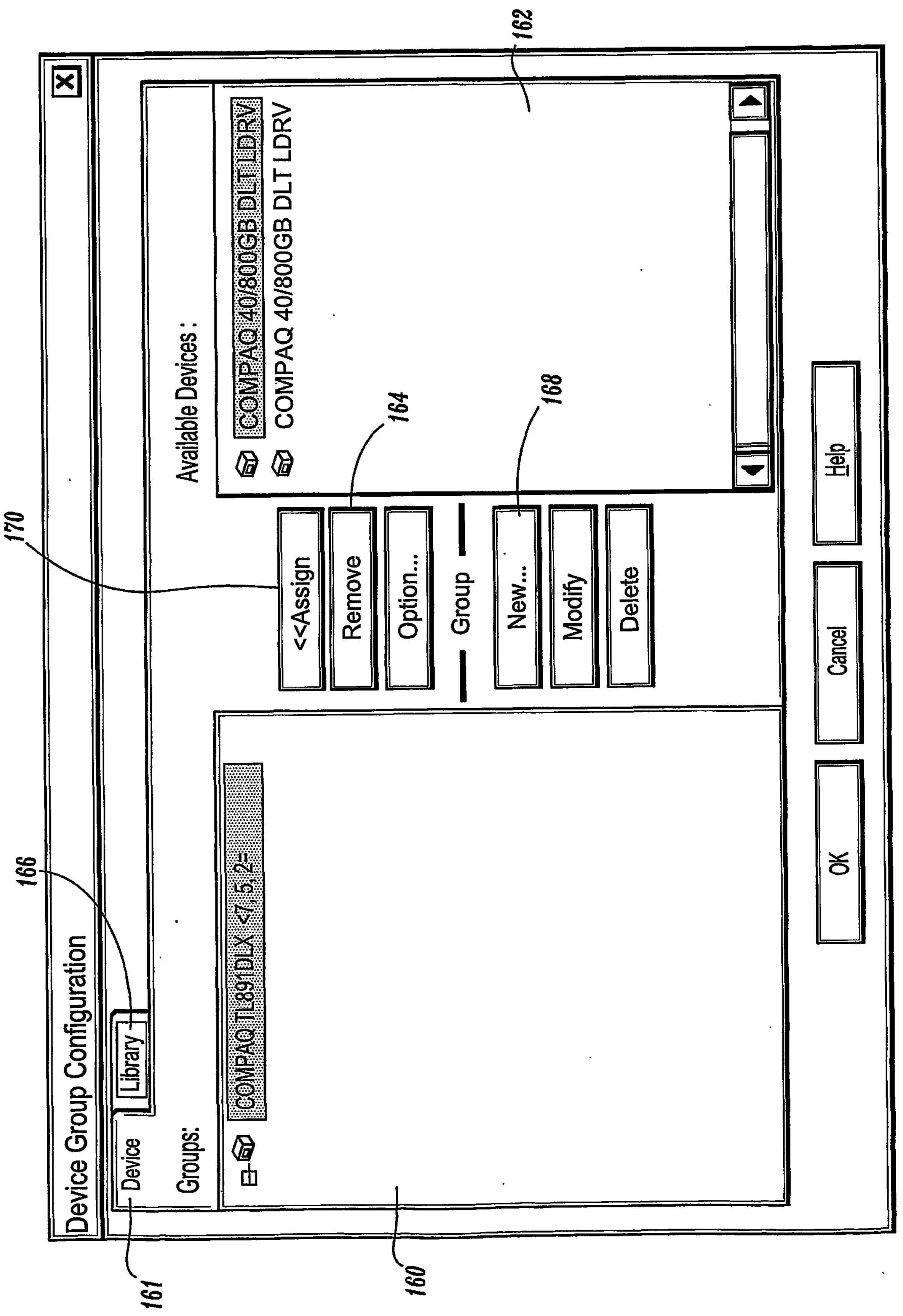

Device Group Configuration
Device
Library
Groups:
COMPAQ TL89'DLX <7, 5, 2>
Available Devices :
COMPAQ 40/800GB DLT LDRV
COMPAQ 40/800GB DLT LDRV
<<Assign
Remove
Option...
Group
New...
Modify
Delete
OK
Cancel
Help
160
161
162
164
166
168
170

FIG. 22

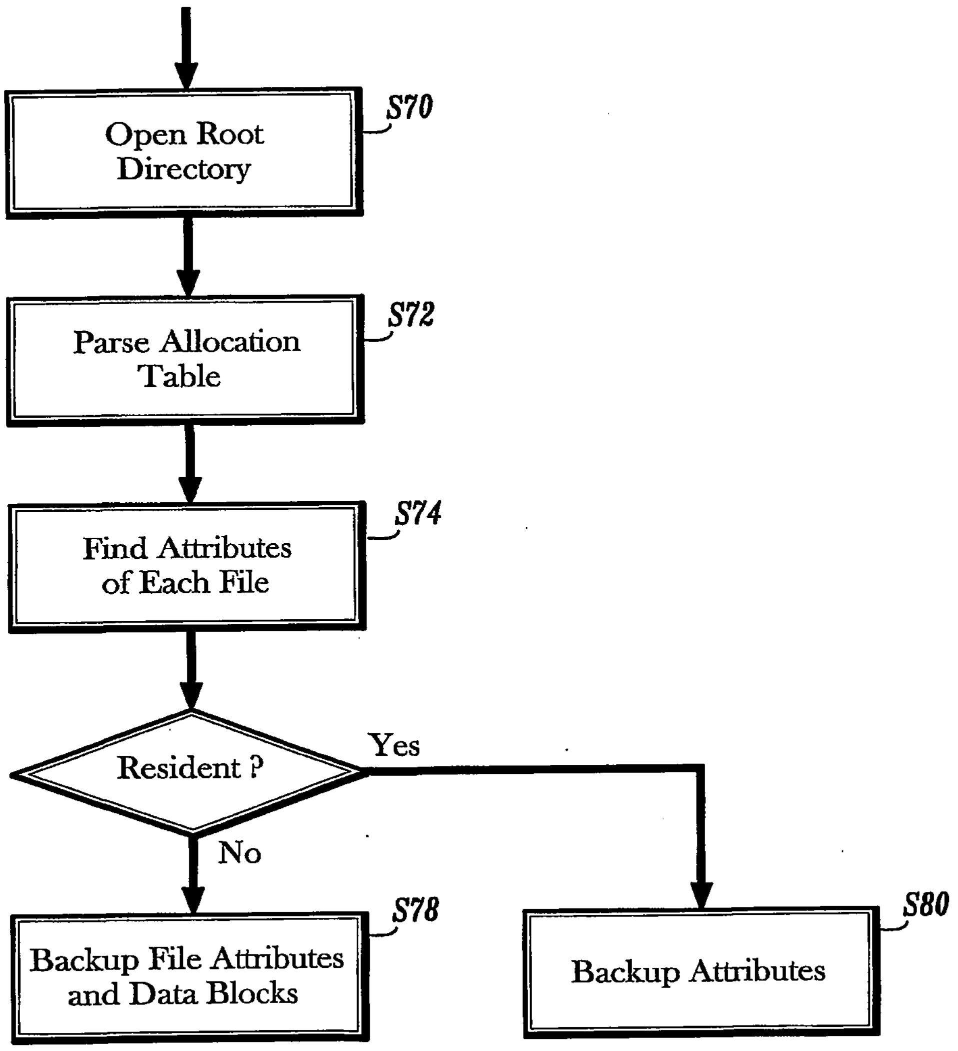

Open Root Directory
S70
Parse Allocation Table
S72
Find Attributes of Each File
S74
Resident ?
Yes
No
Backup File Attributes and Data Blocks
S78
Backup Attributes
S80

FIG. 23

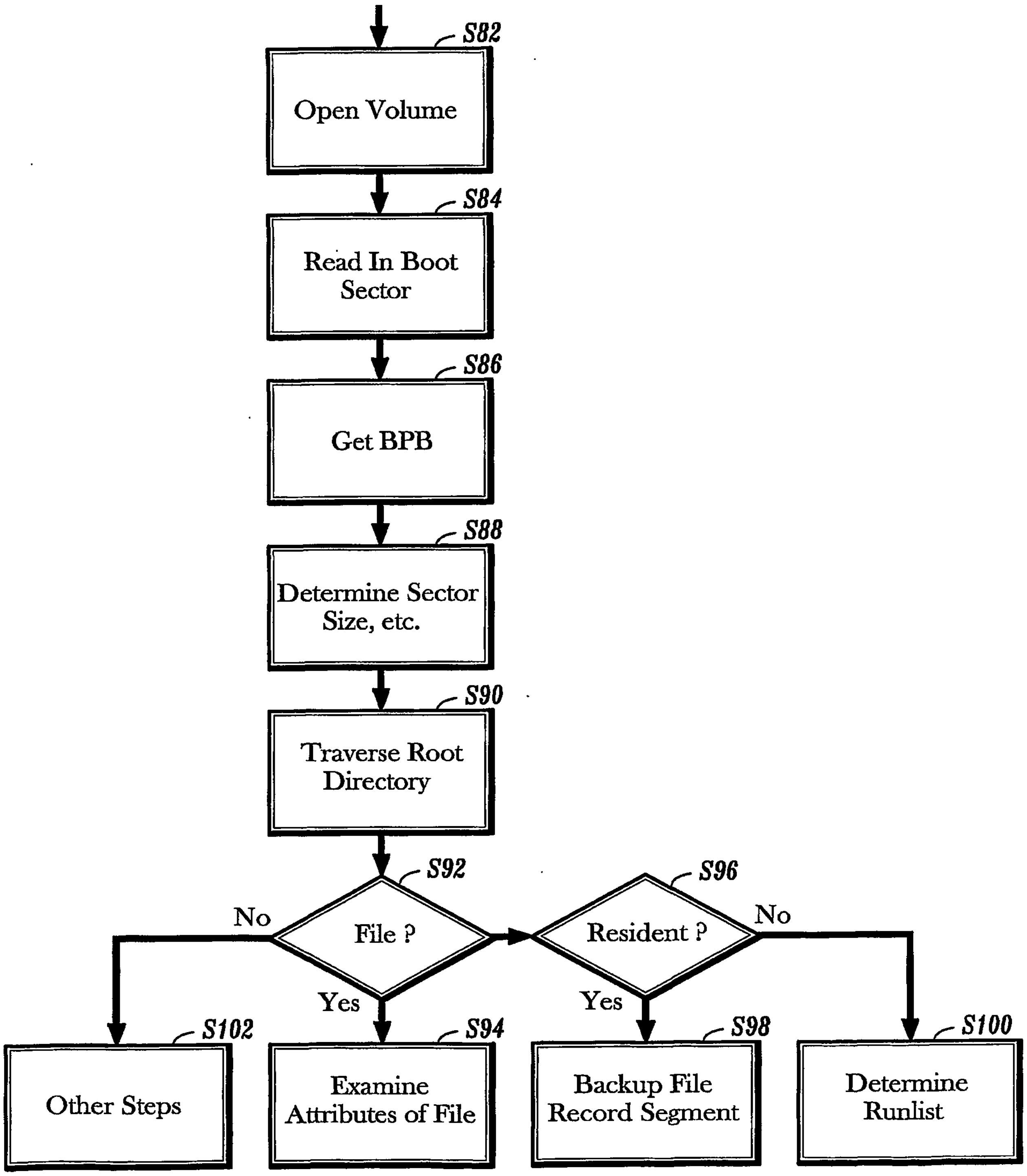

S82
Open Volume
S84
Read In Boot Sector
S86
Get BPB
S88
Determine Sector Size, etc.
S90
Traverse Root Directory
S92
File ?
No
Yes
S96
Resident ?
No
Yes
S102
Other Steps
S94
Examine Attributes of File
S98
Backup File Record Segment
S100
Determine Runlist

FIG. 24

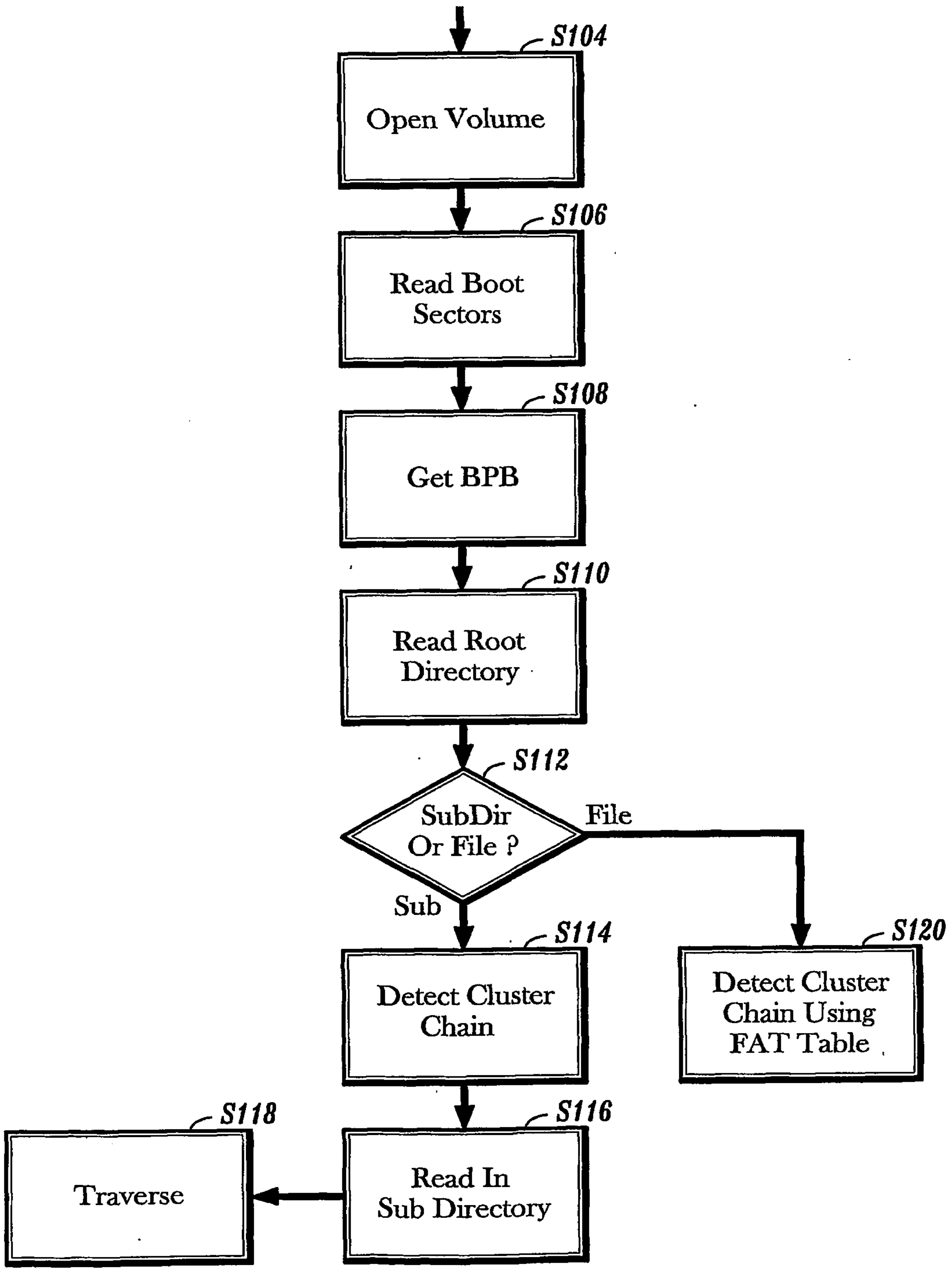

S104
Open Volume
S106
Read Boot Sectors
S108
Get BPB
S110
Read Root Directory
S112
SubDir Or File ?
File
Sub
S114
Detect Cluster Chain
S120
Detect Cluster Chain Using FAT Table
S118
Traverse
S116
Read In Sub Directory

FIG. 25